

CCIE Service Provider v5.0 bootcamp workbook

v202106

(c) 2015-2021

Łukasz Bromirski (CCIE #15929 R&S/SP, CCDE #2012::17)
Piotr Jabłoński (CCIE #19476 R&S/SP, CCDE #2012::7)

https://netbrain.zone/

General Table of Contents

Chapter 1: Workbook introduction **12**

Chapter 2: Basic device configuration **14**

Chapter 3: OSPF & ISIS **56**

Chapter 4: Basic BGP **111**

Chapter 5: BGP traffic engineering (TE) **170**

Chapter 6: Advanced BGP **194**

Chapter 7: MPLS Basics **249**

Chapter 8: MPLS Intra-AS L3VPN **284**

Chapter 9: MPLS Inter-AS L3VPN **320**

Chapter 10: MPLS L2VPN: E-LINE **369**

Chapter 11: MPLS L2VPN: E-LAN, E-TREE **396**

Chapter 12: MPLS Traffic Engineering **412**

Chapter 13: MPLS Quality of Service **493**

Chapter 14: Segment Routing **525**

Chapter 15: IPv6 transition **552**

Chapter 16: IP Multicast **588**

Chapter 17: System management tasks **629**

Chapter 18: System and network security **663**

Chapter 19: Network automation in SP networks **706**

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Main Table of Contents

Chapter 1: Workbook introduction 12
Topology overview 13

Chapter 2: Basic device configuration 14
Task 1: Cisco IOS-XE connection 15
Task 2: Cisco IOS XR connection 16
Task 3: IOS-XE and IOS XR configuration capabilities 19
Task 3.1: System archive (IOS-XE) 19
Task 3.2: Configuration rollback (IOS XR) 21
Task 4: IOS XR feature centric configuration 25
Task 5: Other features of IOS XR CLI 26
Task 5.1: Parser regexp matching for selecting interfaces 26
Task 5.2: Configuration replace option 28
Task 5.3: Filtering parser output 29
Task 5.4: Configuration group policies 31
Task 5.5: Commenting configuration 33
Task 5.6: Displaying configuration for different use cases 34
Task 6: IOS XR RPL capabilities 35
Task 6.1: Basic RPL interaction 35
Task 6.2: Editing RPL policies 36
Task 6.3: Checking other RPL options 37
Task 6.4: RPL pass/set/done/drop options 38
Task 6.5: Nested RPL policies 40
Task 6.6: Using RPL parameters 41
Task 6.7: Viewing RPL status 42
Task 7: Traffic filtering and manipulation 43
Task 7.1: Access-lists and object-groups 43
Task 8: Basic IPv4 configuration 45
Task 9: Basic IPv6 configuration 47
Task 9.1: IPv6 address assignment in IOS-XE – EUI64 48
Task 9.2: IPv6 address assignment in IOS XR – EUI64 49
Task 9.3: IPv6 neighbors 50
Task 9.4: IPv6 SLAAC 53
Task 9.5: IPv6 DHCP 54

Chapter 3: OSPF & ISIS 56
Tricky OSPF 56
Topology 1: AS3356 OSPF 57
Task 1: Loopback interface prefix 59
Task 2: Why do prefixes persist? 63
Task 3: Why there is a loop? 67
Task 4: Why ECMP is not equal? 70
AS3356 OSPF 71
Task 1: Migrate OSPF 1 to core 71
Task 2: Inter-area communication 74
Task 3: Scalability and path optimization 78
Task 4: Secure the OSPF domain 86

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

AS3356 OSPFv3 88
Task 1: Configure OSPFv3 areas 88
Task 2: Secure the OSPFv3 domain 93
AS1239 ISIS 94
Topology 1: AS1239 ISIS 94
Task 1: Configure ISIS Level-2 95
Task 2: Secure the ISIS domain 103
AS1239 IP LFA 106
Task 1: AS1239 IP LFA 106

Chapter 4: Basic BGP 111

Task 1: Basic BGP configuration 112
Task 1.1: BGP in AS 100 112
Task 1.1.1: Simple iBGP session 112
Task 1.1.2: BGP router-id selection 114
Task 1.1.3: Advertise network in BGP using 'network' statement 115
Task 1.1.4: Advertise network in BGP using 'redistribute' statement 118
Task 1.1.5: Loopback peering in BGP 121
Task 1.2: ASPLAIN and ASDOT notations in IOS/IOS-XE 123
Task 1.2.1: eBGP session over loopbacks 123
Task 1.2.2: Changing between ASPLAIN and ASDOT 124
Task 1.3: Filtering advertisements in IOS and IOS-XE 125
Task 1.3.1: Using route-maps to filter updates in BGP 125
Task 1.3.2: Using filter-list to filter updates in BGP 127
Task 1.3.3: Reconfigure AS100 back to original configuration 127
Task 1.4: BGP Address Families in IOS and IOS-XE 128
Task 1.4.1: IPv6 unicast session configuration in BGP 129
Task 1.5: Configure routers in AS 3356 131
Task 1.5.1: Simple iBGP session between R1 and R2, R3 and R5 131
Task 1.5.2: BGP router-id selection 134
Task 1.5.3: BGP neighbor status and session options in IOS XR 135
Task 1.5.4: BGP route entry in IOS XR 137
Task 1.5.5: Advertise network in BGP using 'redistribute' statement 138
Task 1.6: Authenticating BGP sessions 141
Task 1.6.1: Using BGP with MD5 with IOS-XE and IOS XR 141
Task 1.6.2: Using BGP with TCP-AO in IOS XR 143
Task 1.7: ASPLAIN and ASDOT notations in IOS/IOS XR 145
Task 1.7.1: Changing between ASPLAIN and ASDOT 145
Task 1.8: Filtering advertisements in IOS XR – using RPL 148
Task 1.8.1: Using RPL with ios-regexp to filter out the prefix while running asdot formatting 149
Task 1.8.2: Rollback SP1R1 to original configuration 152
Task 1.9: BGP Address Families in IOS XR 153
Task 1.9.1: IPv6 unicast session configuration in BGP 153
Task 1.9.2: iBGP full mesh in AS3356 155
Task 1.10: Configure SP1R4 and SP2R4 routers in AS 44.66 159
Task 1.11: Configure confederation in AS 1239 161
Task 1.12: Configure routers in AS 200 166
Task 2: eBGP peering 168
Task 2.1: Configure peering between AS100 and AS3356 168
Task 2.2: Configure peering between AS3356 and AS44.66 168
Task 2.3: Configure peering between AS3356 and AS1239 168
Task 2.4: Configure peering between AS1239 and AS44.66 168
Task 2.5: Configure peering between AS1239 and AS200 168

Chapter 5: BGP traffic engineering (TE) 170

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: Inter-AS Traffic engineering 170
Task 1.1: AS100 Traffic Engineering 170
Task 3.2: AS3356 to AS100 Traffic Engineering 173
Task 3.3: Announcing aggregates in IOS XR and in IOS-XE 179
Task 3.4: AS3356 to AS44.66 Traffic Engineering 181
Task 3.5: AS3356 and AS1239 Traffic Engineering 182
Task 3.6: AS1239 to AS200 Traffic Engineering 183
Task 3.7: AS200 to AS1239 Traffic Engineering 185
Task 4: Intra-AS Traffic engineering 188
Task 4.1: AS3356 Traffic Engineering (with route reflector) 188
Task 4.2: AS1239 Traffic Engineering 190
Task 5: Wrap-up 192
Task 5.1: AS100 setup 192
Task 5.2: AS3356 192
Task 5.3: AS1239 193
Task 5.4: AS200 193
Task 5.5: Inter-AS BGP peerings 193

Chapter 6: Advanced BGP 194

Task 1: Path selection in RR-driven BGP networks 195
Task 1.1: BGP multipath 195
Task 1.2: BGP DMZ link bandwidth on IOS-XE 200
Task 1.3: BGP DMZ link bandwidth on IOS XR 205
Task 1.4: BGP AddPath extension 208
Task 1.5: BGP ORR: Optimal Route Reflection 212
Task 2: Advanced manipulation of path attributes 217
Task 2.1: Accumulated IGP cost attribute 217
Task 2.2: Using local-as BGP feature 219
Task 2.3: Using no-prepend BGP feature 221
Task 2.4: Using replace-as BGP feature 222
Task 2.5: Using dual-as BGP feature 223
Task 2.6: Using allowas-in BGP feature 225
Task 2.7: Using remove-private-as BGP feature 227
Task 2.8: Manipulating communities 230
Task 2.9: Attribute filtering 231
Task 2.10: Selective BGP AD change for prefix 233
Task 3: Real world BGP ISP challenges 235
Task 3.1: BGP route dampening 235
Task 3.2: Selective FIB download – IOS/IOS-XE 237
Task 3.3: Selective FIB download – IOS XR 238
Task 3.4: Multi-instance BGP 239
Task 3.5: BGP multi-session capability 243
Task 3.6: ORF: Optimal Route Filtering 245

Chapter 7: MPLS Basics 249

Task 1: Basic LDP configuration for label distribution 249
Task 1.1: Manually configure LDP protocol on IOS-XE 249
Task 1.2: Manually configure LDP protocol on IOS XR 252
Task 1.3: LDP RID vs transport address 255
Task 1.4: Using MPLS LDP autoconfig on IOS-XE 258
Task 1.5: Using MPLS LDP autoconfig on IOS XR 259

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.6: Control label range for LDP 260
Task 1.7: LDP sync 261
Task 1.8: LDP session protection 262
Task 1.9: LDPv6 or labels for IPv6 prefixes 265
Task 2: Basic MPLS operations 267
Task 2.1: Checking MPLS label forwarding for LDP bindings 267
Task 2.2: Explicit NULL label 270
Task 2.3: Configuring MPLS TTL propagation 271
Task 2.4: Static binding of MPLS labels 272
Task 2.5: LDP security – authentication 276
Task 2.6: LDP security – label filtering 277
Task 2.7: Turning on MPLS in AS1239 and AS200 279
Task 3: Basic MPLS OAM 280
Task 3.1: Turning on MPLS OAM for IOS XR 280
Task 3.2: Turning on MPLS OAM for IOS XE 280
Task 3.3: Using MPLS ping and traceroute on IOS XE 281
Task 3.4. Using MPLS ping and traceroute on IOS XR 282

Chapter 8: MPLS Intra-AS L3VPN 284

Task 1: L3VPN in AS 3356 285
Task 1.1: Enable VPNv4 and connect Customer1 in Site A 285
Task 1.2: Add R1, R3 and R4 to L3VPN 291
Task 1.3: Additional paths with Unique RD 292
Task 1.4: Provide redundancy to site A 295
Task 1.5: Add two additional VRF sites. 298
Task 1.6: Shared services 305
Task 2: MPLS transport for IPv6 308
Task 2.1: 6VPE Service 308
Task 2.2: 6PE Service 313
Task 3: L3VPN in AS 1239 316
Task 3.1: Enable L3VPN 317
Task 3.2: Configure Site D prefixes 317
Task 3.3: Connect VRF cust1 Site D 317
Task 3.4: Connect VRF sitee 317
Task 3.5: Verify prefix advertisement 318

Chapter 9: MPLS Inter-AS L3VPN 320

Task 1: Inter-AS option A 320
Task 1.1: Build inter-AS connectivity 320
Task 1.2: Verification 322
Task 2: Inter-AS option B 324
Task 2.1: Build inter-AS connectivity 324
Task 2.2: Prefix advertisement verification 325
Task 2.3: Traffic engineering 328
Task 3: Inter-AS option C 331
Task 3.1: Build Inter-AS connectivity 331
Task 3.2: Traffic-engineering 333
Task 3.3: The correct BGP next-hop 337
Task 3.4: Continuous LSP 341
Task 4: CSC 346
Task 4.1: New subinterfaces and VRF 347

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.2: End to end LSP 350
Task 4.3: Enable the L3VPN service in AS 100/200 353
Task 5: OSPF as PE-CE protocol 355
Task 5.1: Deploy OSPF 355
Task 5.2: LSA Type manipulation 359
Task 6: EIGRP as PE-CE protocol 363
Task 6.1: Deploy EIGRP 363
Task 6.2: Loop prevention 366
Chapter 10: MPLS L2VPN: E-LINE 369
Task 0: Lab preparations 370
Task 0.1: Continuous LSP 370
Task 1: VPWS 372
Task 1.1: PW with .1Q 372
Task 1.2: PW with QinQ + local bridging 378
Task 1.3: Multisegment PW with a VLAN tag translation 381
Task 1.4: H-VPLS for E-LINE 386
Task 1.5: Point to point EVPN 389
Task 1.6: L2TPv3 393
Chapter 11: MPLS L2VPN: E-LAN, E-TREE 396
Task 1: VPLS 397
Task 1.1: E-LAN BGP signalled VPLS 397
Task 1.2: E-TREE BGP signalled VPLS 402
Task 2: PBB-EVPN 406
Task 2.1: E-LAN BGP signalled PBB-EVPN 406
Chapter 12: MPLS Traffic Engineering 412
Task 1: Basic MPLS TE configuration 413
Task 1.1: Enabling RSVP on interfaces 414
Task 1.2: Interface parameters from RSVP-TE point of view 415
Task 1.3: Building dynamic tunnel 416
Task 1.3.1: Enabling RSVP on all core interfaces 416
Task 1.3.2: Enabling IGP for MPLS Traffic Engineering 416
Task 1.3.3: Enabling MPLS Traffic Engineering on IOS-XE devices 417
Task 1.3.4: Configuring dynamic MPLS TE tunnel 417
Task 1.3.5: IGP cost vs TE cost 422
Task 1.3.6: Announcing TE tunnel via forwarding adjacency 424
Task 1.4: Building static tunnel 426
Task 2: Running MPLS TE in OSPF (AS 3356) 427
Task 2.1: Basic IGP and MPLS TE configuration 427
Task 2.2: Inter-AS MPLS TE Tunnel 429
Task 3: Other MPLS TE features 436
Task 3.1: MPLS TE setup and holding priority 436
Task 3.2: RSVP authentication 440
Task 3.3: Naming MPLS-TE tunnels 441
Task 3.4: Named MPLS-TE tunnels 442
Task 3.5: FlexLSP – bidirectional tunnels 443
Task 3.5.1: Co-routed bidirectional tunnels: autoroute 447
Task 3.5.2: Co-routed bidirectional tunnels: forwarding adjacency 448
Task 4: MPLS TE path protection 449
Task 4.1: Building protected path from SP2R2 to SP2R5 449

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5: MPLS TE link protection 453
Task 5.1: Configuring MPLS TE link protection (NHOP) 453
Task 6: MPLS TE node protection 458
Task 6.1: Configuring MPLS TE node protection (NNHOP) 458
Task 7: MPLS TE advanced path selection 465
Task 7.1: Link affinity with new configuration style - IOS XR 465
Task 7.2: Link affinity with old configuration style - IOS XR 469
Task 7.3: Link affinity with IOS-XE 470
Task 7.4: Shared risk link groups (SRLG) 472
Task 8: MPLS TE auto mesh 479
Task 8.1: Configuring MPLS TE auto-tunnel mesh 479
Task 9: Configuring PE-P MPLS TE tunnels 483
Task 10: MPLS TE load balancing 487
Task 10.1: Configuring MPLS TE load balancing with IOS XE 487
Task 10.2: Configuring MPLS TE load balancing with IOS XR 491

Chapter 13: MPLS Quality of Service 493

Task 1: MPLS QoS DS-TE 493
Task 1.1: DS-TE configuration – RDM model 493
Task 1.2: DS-TE configuration – MAM model 499
Task 1.3: CBTS – IOS only 502
Task 1.4: PBTS - IOS XR only 504
Task 2: Service provider MPLS QoS basics 509
Task 2.1: Simple MQC operations on EXP/TC 510
Task 2.2: Policing of traffic with specific EXP/TC value 511
Task 3: Implementing uniform MPLS QoS model 512
Task 3.1: Checking probes on AS 100 and AS 200 513
Task 3.2: Configuring PE SP1R8 router 515
Task 4: Implementing pipe MPLS QoS model 517
Task 4.1: Configuring PE routers – SP1R7 and SP1R8 517
Task 4.1.1: Explicit null signalling from SP1R7 517
Task 4.1.2: Modifying the EXP values on SP1R7 and SP1R8 518
Task 4.1.3: Queueing on SP1R7 and SP1R8 520
Task 5: Implementing short pipe MPLS QoS model 523
Task 5.1: Configuring PE routers – SP1R7 and SP1R8 523

Chapter 14: Segment Routing 525

Task 1: Segment Routing in OSPF domain of AS3356 525
Task 1.1: Configuring Segment Routing for OSPF on IOS XR 525
Task 1.2: Configuring Segment Routing for OSPF on IOS-XE 528
Task 1.3: Verification 529
Task 1.4: Migrating AS3356 from LDP to SR 530
Task 1.4.1: Making SR preferred over LDP 530
Task 1.4.2: Verification 530
Task 1.4.3: Turning off LDP in AS3356 531
Task 2: Segment Routing in IS-IS domain of AS1239 532
Task 2.1: Configuring Segment Routing for IS-IS on IOS XR 532
Task 2.2: Configuring Segment Routing for IS-IS on IOS-XE 532
Task 2.3: Verification 533
Task 2.4: Migrating AS1239 from LDP to SRs 534
Task 2.4.1: Making SR preferred over LDP 534
Task 2.4.2: Verification 534

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.4.3: Turning off LDP in AS 1239 534
Task 3: Segment Routing TE in IOS XR pre-6.3 535
Task 3.1: Simple SR TE with MPLS data plane with OSPF 535
Task 3.2: Simple SR TE with MPLS data plane with IS-IS 538
Task 4: Segment Routing interworking 540
Task 4.1: Segment Routing Mapping Server feature 540
Task 5: New Segment Routing features 542
Task 5.1: Simple SR TE with MPLS data plane with OSPF 542
Task 5.2: Simple SR TE with MPLS data plane with IS-IS 546
Task 6: Segment Routing additional features 548
Task 6.1: Traffic Collector 548
Task 6.2: Segment Routing MPLS OAM 550
Chapter 15: IPv6 transition 552
Task 1: Tunnelling and translation 553
Task 1.1: VRF-aware NAT44 553
Task 1.2: 6rd 559
Task 1.3: Pseudo DS-Lite 564
Task 1.4: Stateful NAT64 568
Task 1.5: Stateless NAT64 572
Task 1.6: MAP-T 576
Task 1.7: LISP 580
Chapter 16: IP Multicast 588
Task 1: Native Multicast 589
Task 1.1: PIM ASM Static RP 589
Task 1.2: PIM ASM with Static Anycast RP 592
Task 1.3: PIM ASM with Static RP and static IGMP Join 599
Task 1.4: PIM ASM with Auto RP 601
Task 2: mVPN Multicast 603
Task 2.1: Profile 0 - Default MDT, GRE, PIM C-mcast Signaling 603
Task 2.2: Profile 6 - MLDP, In-Band Signaling 613
Task 2.3: Profile 12 - MLDP, P2MP, BGP AD + C-mcast Signaling 619
Task 2.4: Inter-AS mVPN 625
Chapter 17: System management tasks 629
Task 1: Logging functions 630
Task 1.1: Turning on logging 630
Task 1.1.1: Local buffered logging 631
Task 1.1.2: Terminal (monitor) logging 632
Task 1.1.3: Console logging 634
Task 1.1.4: Logging filters 635
Task 1.1.5: Supressing duplicate messages – IOS XR only 636
Task 1.2: Logging to external syslog server 637
Task 1.2.1: Assigning specific prefix/name for logging host 637
Task 1.3: Logging security 639
Task 1.4: SNMP configuration 640
Task 1.4.1: Basic SNMPv2c configuration 640
Task 1.5: RMON configuration 642
Task 2: Debugging network traffic using captures 644
Task 2.1: EPC – IOS-XE 644
Task 3: Network telemetry 648

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.1: Configuring NetFlow in IOS/IOS-XE 648
Task 3.1.1: Defining flow key records 648
Task 3.1.2: Defining flow monitor 649
Task 3.1.3: Defining flow exporter 649
Task 3.1.4: Attaching NetFlow feature to interfaces 650
Task 3.2: BGP policy accounting in IOS/IOS-XE 652
Task 3.3: BGP policy accounting in IOS XR 654
Task 3.4: CEF traffic accounting 656
Task 4: Monitoring network conditions and acting on results 657
Task 4.1: IP SLA in IOS-XE 657
Task 4.2: IP SLA in IOS XR 660
Task 4.3: EEM 661
Task 4.3.1: Basic EEM script 661
Task 4.3.2: More advanced example – network interfaces 661
Task 4.3.3: Building script to bring Lo0 back up again once Track 10 is up 662
Chapter 18: System and network security 663
Task 1: Securing control plane 663
Task 1.1: CoPP 663
Task 1.2: LPTS 666
Task 1.3: BGP security 668
Task 1.3.1: BGP maximum prefix feature on IOS XE 668
Task 1.3.2: BGP VRF route limit on IOS XE 669
Task 1.3.3: BGP maximum prefix feature on IOS XR 670
Task 1.3.4: BGP VRF route limit on IOS XR 671
Task 1.3.5: BGP attribute limits 672
Task 2: Securing management plane 673
Task 2.1: MPP 673
Task 2.1.1: MPP on IOS-XE 673
Task 2.1.2: MPP on IOS XR 673
Task 2.2: Telnet and SSH 675
Task 2.2.1: Telnet and SSH on IOS-XE 675
Task 2.2.2: Telnet and SSH on IOS XR 676
Task 3: Securing forwarding plane 677
Task 3.1: uRPF 677
Task 3.1.1: uRPF on IOS XR 677
Task 3.1.2: uRPF on IOS/IOS-XE 678
Task 3.2: BGP RPKI with RPKI server 679
Task 3.2.1: Configure RPKI with IOS-XE connection 680
Task 3.2.2: Prefix injection towards AS 3356 682
Task 3.2.3: Running RPKI with IOS XR connection 683
Task 3.2.4: RPKI signalling in IOS-XE 688
Task 3.2.5: RPKI policies in IOS-XE 689
Task 3.2.6: RPKI policies with IOS XR 691
Task 4: Deploying SP-wide protection mechanisms 693
Task 4.1: BGP blackholing 693
Task 4.1.1: Configuring trigger router 693
Task 4.1.2: Configuring edge router 696
Task 4.1.3: “Starting the DDoS” 697
Task 4.2: BGP FlowSpec 698
Task 4.2.1: Configuring trigger router (FlowSpec server) 698
Task 4.2.1: Configuring edge router (FlowSpec client) 699
Task 4.3: QoS Policy Propagation via BGP (QPPB) 701
Task 4.3.1: Configuring trigger router 701
Task 5: Other system-related options 705
Task 5.1: Benchmarking/testing link between routers 705

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 19: Network automation in SP networks 706
Task 1: Basic Ansible concepts 708
Task 1.1: Installing & verifying Ansible 708
Task 1.2: Initial Ansible configuration 708
Task 1.3: First Ansible playbook 710
Task 1.4: Running Ansible playbooks in debug mode 712
Task 2: Advanced Ansible concepts 714
Task 2.1: Introducing configuration changes using Ansible 714
Task 2.2: Monitoring reachability using Ansible 716
Task 3: Basic NSO concepts 718
Task 3.1: Introducing NSO 718
Task 3.2: Staring your NSO instance 720
Task 3.2.1: Adding NEDs to our NSO instance 721
Task 3.2.2: Working with NSO CLI 722
Task 3.3: Saving and loading device information 726
Task 4: Advanced NSO concepts 727
Task 4.1: Cisco NSO template basics 727

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 1: Workbook introduction

This workbook is part of the CCIE SP v4.1 bootcamp we're also conducting. It aims to help you prepare and pass CCIE Service Provider v4.1 practical exam. While it can be used as a stand alone self-study guide, we highly recommend joining our bootcamp, in person or online, to get expert guidance from us.

This is to say, that exercises and tasks provided in the workbook can be treated as self-study, but not everything covered during bootcamp has workbook representation, and some of the topics are covered only on a very high (yet, technical) level – during bootcamp, we take everything apart and investigate all zeros and ones behind inner workings.

Also, occasionally workbook contains references to external materials, but most important study resources are gathered in one HTML file with sections per session – the file that's on the github repository here:

https://github.com/lukasz-bromirski/netdesign.zone

If you want to join the bootcamp, you'll receive:

- 10 WebEx sessions covering all of the blueprint; WebEx sessions will be recorded and available up to 30 days after bootcamp ends
- latest revision of this workbook along with updates until the exam revision changes
- access to your own pod, available 24/7 during the bootcamp and up to month after the bootcamp
- additional PDF with per-chapter topologies, ready to print if you need it to help during various configuration and reconfiguration tasks
- two additional scenarios simulating real CCIE SP exam approach; first three students that will handle those scenarios will get scores and will get feedback on gaps to cover before attempting real exam

Please contact us using below information if you're interested:

https://netbrain.zone/#contact-1

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

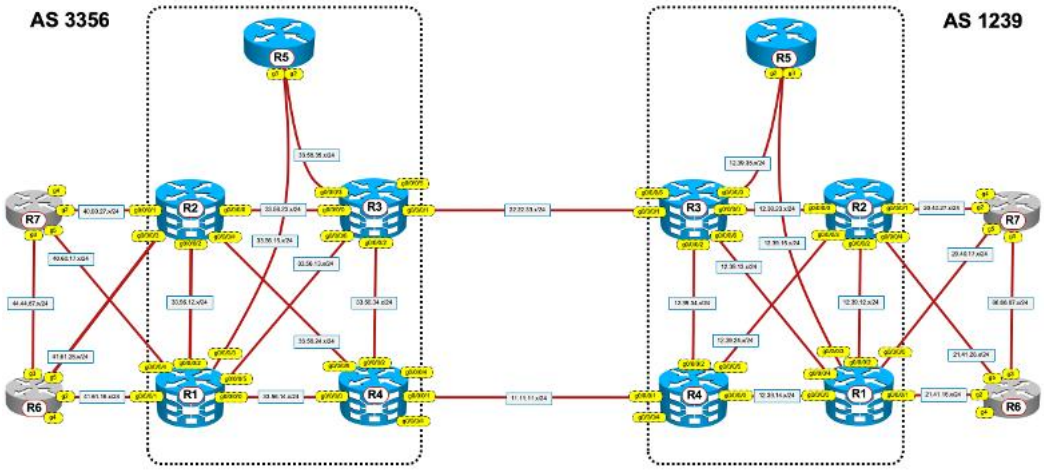
Topology overview

During the whole CCIE SP Zero to Hero class you'll be using below topology. Some connections between devices may be up and used, while some of them **may be shut down for the specific scenario**. Please observe initial topology and configuration before starting the lab exercises.

Topology consists of two fictious Service Providers along with interconnection between them and some customer (CE) routers:

AS 1239
using 12.39.x.x/16 IPv4 addressing space for infrastructure, 12.39.R.R/32 for loopbacks

AS 3356
using 33.56.x.x/16 IPv4 addressing space for infrastructure, 33.56.R.R/32 for loopbacks



Picture 1. General lab topology

Please note that we're using following notation:

- **R** – means router number, denoted by 'RX' on the topology diagram, where 'X' is number from 1 to 9
- **RxRy** – in the prefix or subnet specification, means subnet between router number X and Y, for example if the prefix is specified as 12.39.RxRy.0/24 and the connection is linking R1 to R7, the prefix will be 12.39.17.0/24.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 2: Basic device configuration

You'll get familiar with two different network operating systems (NOS) during this introductory lab.

This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch02.virl1.virl`
- **For VIRL²/CML²:** `sp.bootcamp.ch02.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch02.unl`

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: Cisco IOS-XE connection

Connect using console connection to SP1R5 device, which is Cisco CSR 1000v device running IOS-XE. The same operating system runs on physical devices like Cisco ISR 4000 series or Cisco ASR 1000 series.

You'll see unprivileged mode. The enable password is '**cisco**':

```
SP1-R5-RR>en
Password:

SP1-R5-RR#sh ver | i IOS
Cisco IOS XE Software, Version 16.04.01
Cisco IOS Software [Everest], CSR1000V Software (X86_64_LINUX_IOSD-UNIVERSALK9-M),
Version 16.4.1, RELEASE SOFTWARE (fc2)
Cisco IOS-XE software, Copyright (c) 2005-2016 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
ROM: IOS-XE ROMMON
```

One of the great additions of the newer IOS and IOS XE devices that can be useful in the real lab is enabling full parser. This gives you some additional capabilities known from UNIX world:

```
SP1-R5-RR#sh ver | grep IOS
                 ^
% Invalid input detected at '^' marker.

SP1-R5-RR#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SP1-R5-RR(config)#shell processing full
SP1-R5-RR(config)#^Z

SP1-R5-RR#sh ver | grep IOS
Cisco IOS XE Software, Version 16.04.01
Cisco IOS Software [Everest], CSR1000V Software (X86_64_LINUX_IOSD-UNIVERSALK9-M),
Version 16.4.1, RELEASE SOFTWARE (fc2)
Cisco IOS-XE software, Copyright (c) 2005-2016 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
ROM: IOS-XE ROMMON
```



This feature, while useful (full shell processing) can break things in the configuration – including special characters that are perfectly acceptable for IOS in names (like '>' often used in route-maps) or EEM scripts ('?' for matching). Please use with caution!

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Cisco IOS XR connection

Connect using console connection to **SP1R1** device, which is Cisco XRv device running IOS XR. The same operating system runs on physical devices like Cisco CRS systems, NCS systems and ASR 9000 systems.

Because of the way EVE-NG boots XRv nodes, despite configuration present the node won't see root-username configured. This in effect will lead it to ask for the first time for each loaded lab and node following clean-boot question:

```
!!!!!! NO root-system username is configured. Need to configure root-system username.
!!!

   --- Administrative User Dialog ---

 Enter root-system username:
```

Our proposal is to configure by default xvr user with xvr password:

```
Enter root-system username: xvr
Enter secret: xvr
Enter secret again: xvr
```

After you pass this point for each XRv node, you'll be able to then login either using this account, or any of the built-in accounts – **cisco**/**cisco** or **admin**/**admin**:

```
IMPORTANT:  READ CAREFULLY
[..]

Please login with any configured user/password, or cisco/cisco


User Access Verification

Username: cisco
Password:

RP/0/0/CPU0:SP1R1#sh ver | i IOS
Wed Jan  1  16:49:29.473 UTC
Cisco IOS XR Software, Version 6.1.2[Default]
cisco IOS XRv Series (Pentium Celeron Stepping 3) processor with 3169791K bytes of
memory.
IOS XRv Chassis

RP/0/0/CPU0:SP1R1#sh ver
Wed Jan  1  16:49:42.232 UTC

Cisco IOS XR Software, Version 6.1.2[Default]
Copyright (c) 2016 by Cisco Systems, Inc.

ROM: GRUB, Version 1.99(0), DEV RELEASE

SP1R1 uptime is 35 minutes
System image file is "bootflash:disk0/xrvr-os-mbi-6.1.2/mbixrvr-rp.vm"

cisco IOS XRv Series (Pentium Celeron Stepping 3) processor with 3169791K bytes of
memory.
Pentium Celeron Stepping 3 processor at 2533MHz, Revision 2.174
IOS XRv Chassis

1 Management Ethernet
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
4 GigabitEthernet
97070k bytes of non-volatile configuration memory.
866M bytes of hard disk.
2321392k bytes of disk0: (Sector size 512 bytes).

Configuration register on node 0/0/CPU0 is 0x2102
Boot device on node 0/0/CPU0 is disk0:
Package active on node 0/0/CPU0:
iosxr-infra, V 6.1.2[Default], Cisco Systems, at disk0:iosxr-infra-6.1.2
    Built on Fri Nov 11 11:32:02 UTC 2016
    By iox-lnx-009 in /auto/srcarchive11/production/6.1.2/xrvr/workspace for pie

[…]
```

After playing up with the console try to reconfigure interface connecting SP1R1 to SP1R5 for IPv6 protocol:

```
RP/0/0/CPU0:SP1R1#sh running-config interface gigabitEthernet 0/0/0/3
interface GigabitEthernet0/0/0/3
 description to SP1R5
 ipv4 address 33.56.15.1 255.255.255.0
!
```

Configure the 2001:db8::15:R/64 subnet on the interface:

```
RP/0/0/CPU0:SP1R1(config)#int gi0/0/0/3
RP/0/0/CPU0:SP1R1(config-if)#ipv6 address 2001:db8::15:1/64
RP/0/0/CPU0:SP1R1(config-if)#commit
RP/0/0/CPU0:SP1R1(config-if)#
RP/0/0/CPU0:SP1R1#sh running-config interface gigabitEthernet 0/0/0/3
interface GigabitEthernet0/0/0/3
 description to SP1R5
 ipv4 address 33.56.15.1 255.255.255.0
 ipv6 address 2001:db8::15:1/64
!
```

For the test, configure same subnet on the SP1R5 IOS XE device:

```
SP1-R5-RR#sh run int gi3
Building configuration...

Current configuration : 141 bytes
!
interface GigabitEthernet3
 description to SP1R1
 ip address 33.56.15.5 255.255.255.0
 negotiation auto
 no mop enabled
 no mop sysid
end

SP1-R5-RR#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SP1-R5-RR(config)#int gi3
SP1-R5-RR(config-if)#ipv6 address 2001:db8::15:5/64
SP1-R5-RR(config-if)#^Z
```

And test for basic connectivity over IPv6:

```
SP1-R5-RR#ping 2001:db8::15:1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8::15:1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
SP1-R5-RR#
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Once the IPv6 neighbor discovery finds a mapping, you can also play with different options on IOS XE and IOS XR to check the contents of IPv6 neighbor cache:

```
RP/0/0/CPU0:SP1R1#show ipv6 neighbors
IPv6 Address                Age Link-layer Addr State   Interface   Location
2001:db8::15:5               81   fa16.3e54.e7e1 REACH  Gi0/0/0/3   0/0/CPU0
fe80::f816:3eff:fe54:e7e1    83   fa16.3e54.e7e1 REACH  Gi0/0/0/3   0/0/CPU0
[Mcast adjacency]             -   0000.0000.0000 REACH  Gi0/0/0/3   0/0/CPU0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3: IOS-XE and IOS XR configuration capabilities

Task 3.1: System archive (IOS-XE)

System archive is IOS and IOS-XE feature that keeps the configs archive and can be used to get the diffs and changes.

Let's setup system archive on flash in arch directory on flash. Current config will be automatically saved each 5 minutes, and on each saving of configuration by operator.

```
CEL6(config)#archive
CEL6(config-archive)#path flash:/arch
CEL6(config-archive)#maximum 10
CEL6(config-archive)#time-period 5
CEL6(config-archive)#write-memory
```

The current config of the archive feature and status can be checked with:

```
CEL6#sh archive
The maximum archive configurations allowed is 10.
There are currently 1 archive configurations saved.
The next archive file will be named flash:/arch-<timestamp>-1
 Archive #  Name
   1        flash:/arch-Jan-01-20-54-34.550-0 <- Most Recent
   2
   3
   4
   5
   6
   7
   8
   9
   10
```

Let's now delete Lo0 interface from config:

```
CEL6(config)#no int lo0
```

Now, there will be next entry in archive:

```
CEL6#sh arch
The maximum archive configurations allowed is 10.
There are currently 2 archive configurations saved.
The next archive file will be named flash:/arch-<timestamp>-2
 Archive #  Name
   1        flash:/arch-Jan-01-20-54-34.550-0
   2        flash:/arch-Jan-01-20-57-39.083-1 <- Most Recent
   3
   4
   5
   6
   7
   8
   9
   10
```

We can now check the diff from the previous configuration:

```
CEL6#sh archive config incremental-diffs flash:arch-Jan-01-20-54-34.550-0
!List of Commands:
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
interface Loopback0
 description Loopback
 ip address 172.16.6.6 255.255.255.255
 ip ospf 100 area 0
 ipv6 address 2001:44:66:CC::6/128
 ipv6 ospf 100 area 0
 ipv6 ospf prefix-suppression disable
router ospf 100
 passive-interface Loopback0
router bgp 100
 neighbor 2001:44:66:CC::7 update-source Loopback0
 neighbor 172.16.7.7 update-source Loopback0
route-map CONN->BGP permit 10
 match interface Loopback0
end
```

…and apply that diff to current running config:

```
CEL6#configure replace flash:arch-Jan-01-20-54-34.550-0 list
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: y
!Pass 1
!List of Rollback Commands:
archive
 no write-memory
interface Loopback0
 description Loopback
 ip address 172.16.6.6 255.255.255.255
 ip ospf 100 area 0
 ipv6 address 2001:44:66:CC::6/128
 ipv6 ospf 100 area 0
 ipv6 ospf prefix-suppression disable
router ospf 100
 passive-interface Loopback0
router bgp 100
 neighbor 2001:44:66:CC::7 update-source Loopback0
 neighbor 172.16.7.7 update-source Loopback0
route-map CONN->BGP permit 10
 match interface Loopback0
end

Total number of passes: 1
Rollback Done

*Jan 01 21:00:06.922: Rollback:Acquired Configuration lock.
*Jan 01 21:00:06.922: %SYS-5-CONFIG_R: Config Replace is Done
*Jan 01 21:00:09.023: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed
state to up
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2: Configuration rollback (IOS XR)

IOS XR introduced different paradigm of configuring device – instead of interactive configuration, IOS XR enables you to first configure and then commit changes.

Let's try how it works on the virtualized device by changing the domain name (as one example):

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#domain name netdesign.zone
```

…checking the candidate config:

```
RP/0/0/CPU0:SP1R1(config)#sh config
Building configuration...
!! IOS XR Configuration 6.1.2
domain name netdesign.zone
end
```

…and finally committing the config with both assigning label and some comment:

```
RP/0/0/CPU0:SP1R1(config)#commit ?
  best-effort   Commit the configuration changes via best-effort operation
  comment       Assign a comment to this commit
  confirmed     Rollback this commit unless there is a confirming commit
  force         Override the memory checks
  label         Assign a label to this commit
  replace       Replace the contents of running configuration
  save-running  Save running configuration to a file
  show-error    Displays commit failures immediately
  <cr>          Commit the configuration changes via pseudo-atomic operation

RP/0/0/CPU0:SP1R1(config)#commit label DOMAIN comment Changed domain name
```

To check the commit list (along with labels) use the '**show config commit list**' command:

```
RP/0/0/CPU0:SP1R1#show configuration commit list
SNo. Label/ID     User      Line            Client      Time Stamp
~~~~ ~~~~~~~~     ~~~~      ~~~~            ~~~~~~      ~~~~~~~~~~
1    DOMAIN       cisco     con0_0_CPU0     CLI         Wed Jan  1  23:28:27 2017
2    1000000002   cisco     con0_0_CPU0     CLI         Wed Jan  1  16:57:56 2017
3    1000000001   CVAC      UNKNOWN         CLI         Wed Jan  1  16:19:46 2017
```

The '**show config commit list detail**' will also display comments per commit:

```
RP/0/0/CPU0:SP1R1#show configuration commit list detail

   1) CommitId: 1000000003                Label: DOMAIN
      UserId:   cisco                     Line:  con0_0_CPU0
      Client:   CLI                       Time:  Wed Jan  1 23:28:27 2017
      Comment:  Changed domain configuration
```

If there was an mistake in the configuration and you want to rollback the configuration, it's also very easy:

```
RP/0/0/CPU0:SP1R1#rollback configuration ?
  DOMAIN      1000000003
  last        Rollback last <n> commits made
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  to           Rollback up to (and including) a specific commit
  to-exclude   Rollback up to (and excluding) a specific commit
  1000000003   Commit ID
  1000000002   Commit ID
  1000000001   Commit ID
RP/0/0/CPU0:SP1R1#rollback configuration DOMAIN
Loading Rollback Changes.
Loaded Rollback Changes in 1 sec
Committing.
1 items committed in 1 sec (0)items/sec
Updating.
Updated Commit database in 1 sec
Configuration successfully rolled back commit 'DOMAIN'.
RP/0/0/CPU0:SP1R1#
```

And we're back with the domain name '**netdesign.zone**':

```
RP/0/0/CPU0:SP1R1#sh running-config | i domain
Building configuration...
domain name netdesign.zone
domain lookup disable
```

You can also, while still in configuration mode execute a '**clear**' command, that will delete all the candidate changes in the current session. Like this:

```
RP/0/0/CPU0:SP1R1(config)#int lo50600
RP/0/0/CPU0:SP1R1(config-if)#ipv4 address 172.16.10.1/24
RP/0/0/CPU0:SP1R1(config-if)#ipv6 address 2001:db8::10:1/64
RP/0/0/CPU0:SP1R1(config-if)#exit
RP/0/0/CPU0:SP1R1(config)#sh config
Building configuration...
!! IOS XR Configuration 6.1.2
interface Loopback50600
 ipv4 address 172.16.10.1 255.255.255.0
 ipv6 address 2001:db8::10:1/64
!
end

RP/0/0/CPU0:SP1R1(config)#clear
RP/0/0/CPU0:SP1R1(config)#sh config
Building configuration...
!! IOS XR Configuration 6.1.2
end
```

To avoid situations, where the commit anyway does cause problems, like for example disconnection from remote router, you can always use '**commit confirmed**' with amount of time configuration will be applied, but rolled back if not confirmed within the specified timeframe. In example for 30 seconds:

```
RP/0/0/CPU0:SP1R1(config)#commit confirmed ?
  <30-65535>  Seconds until rollback unless there is a confirming commit
  minutes     Specify the rollback timer in the minutes
  show-error  Displays commit failures immediately
  <cr>        Commit the configuration changes via pseudo-atomic operation
RP/0/0/CPU0:SP1R1(config)#commit confirmed 30
RP/0/0/CPU0:SP1R1(config)#do sh int lo50600
Loopback50600 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Loopback interface(s)
  Internet address is 172.16.10.1/24
  MTU 1500 bytes, BW 0 Kbit
     reliability Unknown, txload Unknown, rxload Unknown
  Encapsulation Loopback,  loopback not set,
  Last link flapped 00:00:29
  Last input Unknown, output Unknown
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Last clearing of "show interface" counters Unknown
  Input/output data rate is disabled.

[ wait a moment ]

RP/0/0/CPU0:SP1R1(config)#do show clock
Sat Jan  1 09:07:40.723 UTC
RP/0/0/CPU0:SP1R1(config)#do sh int lo50600
Interface not found (Loopback50600)
```

So the configuration was rolled back. We can check what was the change pending and not confirmed in time:

```
RP/0/0/CPU0:SP1R1(config)#do show configuration rollback changes last 1
Building configuration...
!! IOS XR Configuration 6.1.2
interface Loopback50600
 ipv4 address 172.16.10.1 255.255.255.0
!
end
```

If something goes wrong while trying to commit the changes, You're informed about that:

```
RP/0/0/CPU0:SP1R2(config)#commit
% Failed to commit one or more configuration items during a pseudo-atomic operation. All
changes made have been reverted. Please issue 'show configuration failed [inheritance]'
from this session to view the errors
```

As instructed, please use '**show configuration failed**':

```
RP/0/0/CPU0:SP1R2(config)#show configuration failed
!! SEMANTIC ERRORS: This configuration was rejected by
!! the system due to semantic errors. The individual
!! errors with each failed configuration command can be
!! found below.
[…]
!
!!% 'policy-lib' detected the 'resource not available' condition 'Null-object'
end
```

You can check candidate change waiting to be committed:

```
RP/0/0/CPU0:SP1R2(config)#sh commit changes diff
Building configuration...
!! IOS XR Configuration 6.1.3
   !
+  class-map type traffic match-any PBF
+   match access-group ipv4 TEST-ACL
+   end-class-map
+  !
   !
+  policy-map type pbr POLICY
+   class type traffic PBF
+    set destination-address ipv4 4.4.4.4
+    transmit
+   !
+   class type traffic class-default
+   !
+   class type none class-default
+    drop
+   !
+   end-policy-map
+  !
end
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

…or the shortcut of above command without ‘diff’ stanza (‘-‘ and ‘+’):

```
RP/0/0/CPU0:SP1R2(config)#show
Building configuration...
!! IOS XR Configuration 6.1.3
  class-map type traffic match-any PBF
   match access-group ipv4 TEST-ACL
   end-class-map
 !
 !
 policy-map type pbr POLICY
  class type traffic PBF
    set destination-address ipv4 4.4.4.4
     transmit
  !
 class type traffic class-default
  !
 class type none class-default
   drop
  !
 end-policy-map
end
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4: IOS XR feature centric configuration

In traditional IOS and IOS XE, to configure static routing you'd execute:

```
CEL6#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
CEL6(config)#ip route 66.66.66.0 255.255.255.0 3.4.5.6
```

In IOS XR configuration is done on the feature level – so in this example, it's '**routing**' section.

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#router static
RP/0/0/CPU0:SP1R1(config-static)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-static-afi)#66.66.66.0/24 3.4.5.6
RP/0/0/CPU0:SP1R1(config-static-afi)#sh config
Building configuration...
!! IOS XR Configuration 6.1.2
router static
 address-family ipv4 unicast
  66.66.66.0/24 3.4.5.6
 !
!
end
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5: Other features of IOS XR CLI

Task 5.1: Parser regexp matching for selecting interfaces

One of the useful features is to use wildcards in the commands to mask some output that's not needed at the moment. For example, you can use wildcard to filter out specific type of interface in potentially lengthy output:

```
RP/0/0/CPU0:SP1R1#sh int brief

               Intf       Intf        LineP              Encap  MTU         BW
               Name       State       State               Type (byte)     (Kbps)
--------------------------------------------------------------------------------
                Lo0          up          up           Loopback  1500           0
                Nu0          up          up               Null  1500           0
       Mg0/0/CPU0/0          up          up               ARPA  1514     1000000
          Gi0/0/0/0          up          up               ARPA  1514     1000000
          Gi0/0/0/1          up          up               ARPA  1514     1000000
          Gi0/0/0/2          up          up               ARPA  1514     1000000
          Gi0/0/0/3          up          up               ARPA  1514     1000000
```

Now let's filter out only the Gigabit Ethernet interfaces:

```
RP/0/0/CPU0:SP1R1#sh int Gi* brief

               Intf       Intf        LineP              Encap  MTU         BW
               Name       State       State               Type (byte)     (Kbps)
--------------------------------------------------------------------------------
          Gi0/0/0/0          up          up               ARPA  1514     1000000
          Gi0/0/0/1          up          up               ARPA  1514     1000000
          Gi0/0/0/2          up          up               ARPA  1514     1000000
          Gi0/0/0/3          up          up               ARPA  1514     1000000
```

The same can be applied to checking configuration:

```
RP/0/0/CPU0:SP1R2#show running-config interface gigabitEthernet *
interface GigabitEthernet0/0/0/0
 description to SP1R3
 ipv4 address 33.56.23.2 255.255.255.0
 ipv6 address fe80::2 link-local
 ipv6 address 2001:33:56:23::2/64
!
interface GigabitEthernet0/0/0/1
 description to CEL7
 ipv4 address 40.60.27.2 255.255.255.0
 ipv6 address fe80::2 link-local
 ipv6 address 2001:40:60:27::2/64
!
interface GigabitEthernet0/0/0/2
 description to SP1R1
 ipv4 address 33.56.12.2 255.255.255.0
 ipv6 address fe80::2 link-local
 ipv6 address 2001:33:56:12::2/64
!
interface GigabitEthernet0/0/0/3
 description to CEL6
 ipv4 address 41.61.26.2 255.255.255.0
 ipv6 address fe80::2 link-local
 ipv6 address 2001:41:61:26::2/64
 shutdown
!
interface GigabitEthernet0/0/0/4
 description to SP1R4
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
ipv4 address 33.56.24.2 255.255.255.0
ipv6 address fe80::2 link-local
ipv6 address 2001:33:56:24::2/64
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5.2: Configuration replace option

IOS XR has a powerful replace option in configuration mode. It can be used to move configuration from one part to the other.

Suppose we've made an mistake, and configuration on Gi0/0/0/1 should be moved to Gi0/0/0/3:

```
RP/0/0/CPU0:SP1R2#sh running-config interface gigabitEthernet 0/0/0/1
interface GigabitEthernet0/0/0/1
 description to CEL7
 ipv4 address 40.60.27.2 255.255.255.0
 ipv6 address fe80::2 link-local
 ipv6 address 2001:40:60:27::2/64
!

RP/0/0/CPU0:SP1R2#sh running-config interface gigabitEthernet 0/0/0/3
interface GigabitEthernet0/0/0/3
 description to CEL6
 ipv4 address 41.61.26.2 255.255.255.0
 ipv6 address fe80::2 link-local
 ipv6 address 2001:41:61:26::2/64
!
```

Following option can be used:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#replace ?
  interface  replace configuration for an interface
  pattern    replace a string pattern in configuration
RP/0/0/CPU0:SP1R2(config)#replace interface gig 0/0/0/3 with gig 0/0/0/1
```

The end effect will be:

```
RP/0/0/CPU0:SP1R2(config)#show
Building configuration...
!! IOS XR Configuration 6.1.3
interface GigabitEthernet0/0/0/1
 description to CEL6
 ipv4 address 41.61.26.2 255.255.255.0
 ipv6 address fe80::2 link-local
 ipv6 address 2001:41:61:26::2/64
!
no interface GigabitEthernet0/0/0/3
end
```

The other option is to replace whole substrings – matching regular expression. This can be used to modify large-scale configuration changes.

In our simplistic scenario, we'll modify all interface descriptions from simple 'to XXX' with 'Interface connected to XXX':

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#replace pattern 'to ' with 'Interface connected to '
Loading.
500 bytes parsed in 1 sec (490)bytes/sec
RP/0/0/CPU0:SP1R2(config)#show
Building configuration...
!! IOS XR Configuration 6.1.3
interface GigabitEthernet0/0/0/0
 no description
 description Interface connected to SP1R3
!
interface GigabitEthernet0/0/0/1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
 no description
 description Interface connected to CEL7
!
end
```

Task 5.3: Filtering parser output

Sometimes it's useful to not define route-policy in static, but rather dynamic terms, like for example change specific prefix attribute but per neighbor. Here's While it's a new feature by its own in IOS and IOS-XE, in IOS XR you can directly use utility functions to parse/filter command outputs:

```
RP/0/0/CPU0:SP1R2#show running-config | utility ?
  cut      Cut out selected fields of each line of a file
  egrep    Extended regular expression grep
  fgrep    Fixed string expression grep
  head     Show set of lines/characters from the top of a file
  less     Fixed string pattern matching
  more     Paging Utility More
  script   Launch a script for post processing
  sort     Sort, merge, or sequence-check text files
  tail     Copy the last part of files
  uniq     Report or filter out repeated lines in a file
  wc       Counting lines/words/characters of a file
  xargs    Construct argument list(s) and invoke a program
```

So, for example, to check how many lines has current running-config, you'd do:

```
RP/0/0/CPU0:SP1R2#show running-config | utility wc -l
Building configuration...
      220
```

Each utility has its own help if you don't remember the UNIX syntax:

```
RP/0/0/CPU0:SP1R2#show running-config | utility sort ?
  WORD          Unix commandline option string , maximum length 80
  dict          (-d) Sort in dictionary order
  fieldSep      (-t) Use char value given here as the field separator
  file          File to be sorted
  ignoreblank   (-b) Ignore leading blank characters in field comparisons
  key           (-k) Define keydef to be a sort key
  lowercase     (-f) Fold uppercase letters into lowercase
  merge         (-m) Merge sorted files
  numeric       (-n) Interpret the field as numeric
  outfile       (-o) Measure the quantity of output in lines
  printable     (-i) Ignore all nonprintable characters
  reverse       (-r) Reverse the sort order
  unique        (-u) Unique, suppress all but one line in each set of lines
                 having equal keys
  usage         Unix options supported for sort utility
  |             Output Modifiers
  <cr>          Shows current operating configuration
```

The built-in 'include' and 'exclude' features can be used multiple times which helps in various corner-cases – like this:

```
RP/0/0/CPU0:SP1R2#sh logging | i 0/0 | i PKT_ | i Down
```

Another useful in real-life option is to provide real-time monitoring of interfaces – which sometimes is useful to check traffic levels and various fast-changing characteristics:

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R2#monitor interface gigabitEthernet 0/0/0/0
SP1R2                Monitor Time: 00:00:06          SysUptime: 04:10:43


GigabitEthernet0/0/0/0 is up, line protocol is up
Encapsulation ARPA

Traffic Stats:(2 second rates)                                  Delta
  Input  Packets:                      8789                         0
  Input  pps:                             1
  Input  Bytes:                      795255                         0
  Input  Kbps (rate):                     0                  (   0%)
  Output Packets:                      8287                         1
  Output pps:                             1
  Output Bytes:                      749845                        84
  Output Kbps (rate):                     0                  (   0%)

Errors Stats:
  Input  Total:                           0                         0
  Input  CRC:                             0                         0
  Input  Frame:                           0                         0
  Input  Overrun:                         0                         0
  Output Total:                           0                         0
  Output Underrun:                        0                         0

Quit='q', Freeze='f', Thaw='t', Clear='c', Interface='i',
Next='n', Prev='p'

Brief='b', Detail='d', Protocol(IPv4/IPv6)='r'
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5.4: Configuration group policies

One of the parser capabilities introduced in IOS XR is working on groups, that can apply to various parts of the configuration. This can help you speed up your working on repeatable configuration stanzas within your node configuration – like for example, applying the same set of couple of commands per physical interface, or applying some set of config options per routing instance.

Let's start from something basic – applying description and CDP for every gigabitethernet interface where such group will be applied.

First, we define the group:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#group XR-IFACE
RP/0/0/CPU0:SP1R1(config-GRP)#interface ?
  BVI                 'RegExp': Bridge-Group Virtual Interface
  Bundle-Ether        'RegExp': Aggregated Ethernet interface(s)
  Bundle-POS          'RegExp': Aggregated POS interface(s)
  CEM                 'RegExp': Circuit Emulation interface(s)
  FastEthernet        'RegExp': FastEthernet/IEEE 802.3 interface(s)
  FortyGigE           'RegExp': FortyGigabitEthernet/IEEE 802.3 interface(s)
  GigabitEthernet     'RegExp': GigabitEthernet/IEEE 802.3 interface(s)
  HundredGigE         'RegExp': HundredGigabitEthernet/IEEE 802.3 interface(s)
[…]
  Null                'RegExp': Null interface
  POS                 'RegExp': Packet over SONET/SDH network interface(s)
  PW-Ether            'RegExp': PWHE Ethernet Interface
  PW-IW               'RegExp': PWHE VC11 IP Interworking Interface
  SRP                 'RegExp': SRP interface(s)
  Serial              'RegExp': Serial network interface(s)
  TenGigE             'RegExp': TenGigabitEthernet/IEEE 802.3 interface(s)
  TwentyFiveGigE      'RegExp': TwentyFiveGigabitEthernet/IEEE 802.3 interface
(s)

! You're being asked about matching regular expression, that will be then
! applied to your subconfiguration. We're after GE interfaces, so:

RP/0/0/CPU0:SP1R1(config-GRP)#interface 'GigabitEthernet.*'
RP/0/0/CPU0:SP1R1(config-GRP-if)#description MyTestDescription
RP/0/0/CPU0:SP1R1(config-GRP-if)#mtu 2000
RP/0/0/CPU0:SP1R1(config-GRP-if)#cdp

! we want the group to apply unified description, change MTU to 2000 and turn on CDP

RP/0/0/CPU0:SP1R1(config-GRP-if)#commit
```

Now, it's for the interfaces:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#interface gigabitEthernet ?
  0/0/0/0  GigabitEthernet Interface Instance
  0/0/0/1  GigabitEthernet Interface Instance
  0/0/0/2  GigabitEthernet Interface Instance
[…]
RP/0/0/CPU0:SP1R1(config)#interface gigabitEthernet 0/0/0/0
RP/0/0/CPU0:SP1R1(config-if)#apply-group ?
  XR-IFACE  Group name
  WORD      Group name
RP/0/0/CPU0:SP1R1(config-if)#apply-group XR-IFACE
RP/0/0/CPU0:SP1R1(config-if)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Let's check GE0/0/0/0 and GE0/0/0/1 config:

```
RP/0/0/CPU0:SP1R1#show run int gi0/0/0/0
interface GigabitEthernet0/0/0/0
 apply-group XR-IFACE ! nothing in the configuration differs apart from this entry
 description to SP1R4
 ipv4 address 33.56.14.1 255.255.255.0
 ipv6 address fe80::1 link-local
 ipv6 address 2001:33:56:14::1/64
!
```

And now for the interface 'live' status:

```
RP/0/0/CPU0:SP1R1#sh int gi0/0/0/0
GigabitEthernet0/0/0/0 is up, line protocol is up
  Interface state transitions: 1
  Hardware is GigabitEthernet, address is fa16.3e84.8447 (bia fa16.3e84.8447)
  Description: to SP1R4
  Internet address is 33.56.14.1/24
  MTU 2000 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 1000Mb/s, unknown, link type is force-up
  […]
```

So, let's check how group policies are inherited:

```
RP/0/0/CPU0:SP1R1#sh run int gi0/0/0/0 inheritance
interface GigabitEthernet0/0/0/0
 description to SP1R4
 cdp ! CDP was applied from group
 mtu 2000 ! ...as was MTU
 ipv4 address 33.56.14.1 255.255.255.0
 ipv6 address fe80::1 link-local
 ipv6 address 2001:33:56:14::1/64
!
```

…and for more details what was inherited:

```
RP/0/0/CPU0:SP1R1#sh run int gi0/0/0/0 inheritance detail
interface GigabitEthernet0/0/0/0
 description to SP1R4
 ## Inherited from group XR-IFACE ! and here we have additional verbatim confirmation
 cdp
 ## Inherited from group XR-IFACE
 mtu 2000
 ipv4 address 33.56.14.1 255.255.255.0
 ipv6 address fe80::1 link-local
 ipv6 address 2001:33:56:14::1/64
!
```

Note, that existing entries (like description above) won't be overwritten by group config.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5.5: Commenting configuration

You can add comments to configuration for making sure even some complex constructs will stay understandable.

Let's add comment to following default route:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#router static
RP/0/0/CPU0:SP1R2(config-static)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-static-afi)#! this is Level3 default GW
RP/0/0/CPU0:SP1R2(config-static-afi)#0.0.0.0/0 33.56.5.5
RP/0/0/CPU0:SP1R2(config-static-afi)#commit

RP/0/0/CPU0:SP1R2#sh running-config router static
router static
 address-family ipv4 unicast
  ! this is Level3 default GW
  0.0.0.0/0 33.56.5.5
 !
!
```

Note that comment will be removed if the paired configuration is removed also:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#router static
RP/0/0/CPU0:SP1R2(config-static)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-static-afi)#no 0.0.0.0/0 33.56.5.5
RP/0/0/CPU0:SP1R2(config-static-afi)#commit
RP/0/0/CPU0:SP1R2(config-static-afi)#
RP/0/0/CPU0:SP1R2#sh running-config router static
% No such configuration item(s)
```

You can also remove a specific comment if it's no longer needed by doing '**clear comment**' at the appropriate config level.

Other option of attaching text information to configuration is to attach **description** to specific configuration entries, like that:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#router static
RP/0/0/CPU0:SP1R2(config-static)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-static-afi)#0.0.0.0/0 33.56.5.5 description "This is Level3
route"
RP/0/0/CPU0:SP1R2(config-static-afi)#commit

RP/0/0/CPU0:SP1R2#sh running-config router static
router static
 address-family ipv4 unicast
  0.0.0.0/0 33.56.5.5 description "This is Level3 route"
 !
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5.6: Displaying configuration for different use cases

Say you want to save your device configuration for easy change tracking, and don't want to rely on router disk drive to store only copy of your precious configuration.

There are different ways of saving configs, locally and externally, and you can automate this as well (that was covered above already).

In some cases however, it's good to get the "copy&paste" ready version of your device configuration.

With Cisco IOS XR it's easy, as the configuration is built based on logical models, so what you see as "running-config" is just representation (rendering) of the database in a format that's requested.

By default, configuration is shown as logical ASCII format:

```
RP/0/0/CPU0:SP1R2#sh running-config
Building configuration...
!! IOS XR Configuration 6.1.3
!
hostname SP1R2
service timestamps log datetime msec
service timestamps debug datetime msec
telnet vrf default ipv4 server max-servers 10
telnet vrf Mgmt-intf ipv4 server max-servers 10
domain name netdesign.zone
domain lookup disable
interface Loopback0
 description Loopback
 ipv4 address 33.56.2.2 255.255.255.255
 ipv6 address 2001:33:56:aa::2/128
!
interface MgmtEth0/0/CPU0/0
 shutdown
```

However, you can for example request formal format using suffix 'formal':

```
RP/0/0/CPU0:SP1R2#sh running-config formal
Building configuration...
!! IOS XR Configuration 6.1.3
!
hostname SP1R2
service timestamps log datetime msec
service timestamps debug datetime msec
telnet vrf default ipv4 server max-servers 10
telnet vrf Mgmt-intf ipv4 server max-servers 10
domain name netdesign.zone
domain lookup disable
interface Loopback0
interface Loopback0 description Loopback
interface Loopback0 ipv4 address 33.56.2.2 255.255.255.255
interface Loopback0 ipv6 address 2001:33:56:aa::2/128
interface MgmtEth0/0/CPU0/0 shutdown
```

This form is ready to be copied & pasted, or generally easier to be quickly diff'ed if you're using code tracking/versioning tools like git for example.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 6: IOS XR RPL capabilities

In IOS and IOS-XE, route policies and various routing functions are enabled by either use of feature-specific capabilities (like – filter-list, access-list, distribute-list) or by route-maps (that include table-maps and maps).

While this gives a lot of flexibility in expressing policies, it also makes troubleshooting and feature consistency very hard on large, complex configs.

IOS XR introduces RPL – Route Policy Language. It is more flexible than route-maps, and has capabilities to embed other route-policies, or invoke parameters within its operation.

Task 6.1: Basic RPL interaction

RPL is only available on IOS XR. Let's configure simple route policy, matching specific prefix and then changing its local preference value:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#route-policy CHECK-8
RP/0/0/CPU0:SP1R2(config-rpl)#?
  #-remark          Remark beginning with '#'
  abort             Discard RPL definition and return to top level config
  add               Add offset to the existing value
  apply             Apply a route policy
  delete            Delete route attributes
  done              Accept this route with no further processing
  drop              Reject this route with no further processing
  end-policy        End of route-policy definition
  exit              Exit from this submode
  if                Begin if-statement
  pass              Pass this route for further processing
  prepend           Prepend to BGP AS-path
  remove            Remove all private-as entries
  replace           Replace a sequence of AS numbers with local AS
  set               Set a route attribute
  show              Show partial RPL configuration
  suppress-route    Suppress specific routes when aggregating
  unsuppress-route  Unsuppress specific aggregated routes
  var               Set a var attribute
  <cr>
RP/0/0/CPU0:SP1R2(config-rpl)#if destination-prefix in ( 8.8.8.0/24 ) then
RP/0/0/CPU0:SP1R2(config-rpl-if)#set local-preference 5000
RP/0/0/CPU0:SP1R2(config-rpl-if)#endif
RP/0/0/CPU0:SP1R2(config)#commit
RP/0/0/CPU0:SP1R2(config-rpl)#
```

Now, this only matches one prefix and it's done. What if we want to match multiple prefixes? Let's create prefix-set for RPL:

```
RP/0/0/CPU0:SP1R2(config)#prefix-set ?
  WORD  Prefix setname
RP/0/0/CPU0:SP1R2(config)#prefix-set BAD-GUYS
RP/0/0/CPU0:SP1R2(config-pfx)#8.8.1.0/24,
RP/0/0/CPU0:SP1R2(config-pfx)#8.8.4.0/24,
RP/0/0/CPU0:SP1R2(config-pfx)#8.8.78.0/24
RP/0/0/CPU0:SP1R2(config-pfx)#end-set
RP/0/0/CPU0:SP1R2(config)#commit
```

Notice, that after each prefix if you're willing to continue adding new ones – there's a comma.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 6.2: Editing RPL policies

Now, let's attach the prefix-list to our route-policy. In the old days, that would require checking the current config:

```
RP/0/0/CPU0:SP1R2#sh rpl route-policy CHECK-8
route-policy CHECK-8
  if destination-prefix in (8.8.8.0/24) then
    set local-preference 5000
  endif
end-policy
!
```

And then editing the policy:

```
RP/0/0/CPU0:SP1R2(config)#route-policy CHECK-8
% WARNING: Policy object route-policy CHECK-8' exists! Reconfiguring it via CLI will
replace current definition. Use 'abort to cancel.
```

However, if you try to edit existing policy, you'll simply overwrite it. To edit inplace, in IOS XR there's option to use editor and easier way to edit policies – using '**edit**' option:

```
RP/0/0/CPU0:SP1R2#edit route-policy CHECK-8 ?
  emacs  to use Emacs editor
  nano   to use nano editor
  vim    to use Vim editor
  <cr>
```

We usually stick to the tools we know the best, so the '**vim**' will be platform of choice BUT please use '**nano**' if you don't have intimate knowledge of vi/vim. After you edit/insert the changes, please save the config. End effect should look like this:

```
route-policy CHECK-8
  if destination in BAD-GUYS or (destination-prefix in (8.8.8.0/24)) then
    set local-preference 5000
  endif
end-policy
```

Notice, the 'if' is nested – first we check for prefix matching the prefix-set of BAD-GUYS or 8.8.8.0/24 and if it matches – only then we modify local-preference.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 6.3: Checking other RPL options

Let's now build a policy that will check local-preference and then if it equals some value (be it – greater than default of 100) – modify BGP MED attribute:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#route-policy CHECK-LP
RP/0/0/CPU0:SP1R2(config-rpl)#if local-preference ?
  eq  Numeric equality
  ge  Numeric greater than or equal comparison
  is  Exact match
  le  Numeric less than or equal comparison
RP/0/0/CPU0:SP1R2(config-rpl)#if local-preference ge ?
  <0-4294967295>  32-bit decimal number
  parameter       Identifier specified in the format: '$' followed by [...]
RP/0/0/CPU0:SP1R2(config-rpl)#if local-preference ge 101 then
RP/0/0/CPU0:SP1R2(config-rpl-if)#set med ?
  +               Increment the attribute with specified value
  -               Decrement the attribute by specified value
  <0-4294967295>  32-bit decimal number
  igp-cost        Internal routing protocol cost
  max-reachable   2^32 - 1 (4294967295)
  parameter       Identifier specified in the format: '$' followed by alphanumer
ic characters
RP/0/0/CPU0:SP1R2(config-rpl-if)#set med 500
RP/0/0/CPU0:SP1R2(config-rpl-if)#endif
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
RP/0/0/CPU0:SP1R2(config)#commit
```

Now, we're build a policy that will check both prefix matching specific prefix-set and then specific community – and setting next-hop to null0:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#route-policy CHECK-BAD-DROP
RP/0/0/CPU0:SP1R2(config-rpl)#if destination-prefix in BAD-GUYS and community matches-
any ( ?
  *                 Wildcard (any community or part thereof)
  <0-65535>         16-bit half-community number
  [                 Left bracket to begin range
  abort             Discard RPL definition and return to top level config
  accept-own        Accept-Own (BGP well-known community)
  dfa-regex         DFA style regular expression
  graceful-shutdown  Graceful Shutdown (BGP well-known community)
  internet          Internet (BGP well-known community)
  ios-regex         Traditional IOS style regular expression
  local-AS          Do not send outside local AS (BGP well-known community)
  no-advertise      Do not advertise to any peer (BGP well-known community)
  no-export         Do not export to next AS (BGP well-known community)
  parameter         Identifier specified in the format: '$' followed by alphanu
meric characters
  peeras            Autonomous System Number of the neighbor
  private-as        Match within BGP private AS range [64512..65534]
RP/0/0/CPU0:SP1R2(config-rpl)#if destination-prefix in BAD-GUYS and community matches-
any (64999:666) then
RP/0/0/CPU0:SP1R2(config-rpl-if)#set next-hop discard
RP/0/0/CPU0:SP1R2(config-rpl-if)#endif
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
RP/0/0/CPU0:SP1R2(config)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 6.4: RPL pass/set/done/drop options

We saw simple policies with RPL, all using action '**set**'. There are four actions in RPL however, with the default option for prefix that 'falls through' the entire route-policy of '**drop**' (deny).

- **pass** – processing continues and if no other matches will be dropping, prefix will be accepted
- **set** – processing continues and if no other matches will be dropping – prefix will be accepted
- **done** – processing stops, prefix is accepted
- **drop** – processing stops, prefix is denied/dropped/rejected

Let's build simple policy to demonstrate this capability:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#route-policy IF-ELSE-IF
RP/0/0/CPU0:SP1R2(config-rpl)#if local-preference eq 100 then
RP/0/0/CPU0:SP1R2(config-rpl-if)#drop
RP/0/0/CPU0:SP1R2(config-rpl-if)#else
RP/0/0/CPU0:SP1R2(config-rpl-else)#if local-preference ge 101 then
RP/0/0/CPU0:SP1R2(config-rpl-if-2)#set med 87
RP/0/0/CPU0:SP1R2(config-rpl-if-2)#pass
RP/0/0/CPU0:SP1R2(config-rpl-if-2)#else
RP/0/0/CPU0:SP1R2(config-rpl-else-2)#drop
RP/0/0/CPU0:SP1R2(config-rpl-else-2)#endif
RP/0/0/CPU0:SP1R2(config-rpl-else)#endif
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
RP/0/0/CPU0:SP1R2(config)#commit
```

Now, let's analyse the processing – with prefix entering the policy with LP value of 100:

```
RP/0/0/CPU0:SP1R2#show rpl route-policy IF-ELSE-IF
route-policy IF-ELSE-IF
  if local-preference eq 100 then
    drop ! prefix will be dropped (rejected) here
  else
    if local-preference ge 101 then
      set med 87
      pass
    else
      drop
    endif
  endif
end-policy
!
```

Now, with LP value of 500:

```
RP/0/0/CPU0:SP1R2#show rpl route-policy IF-ELSE-IF
route-policy IF-ELSE-IF
  if local-preference eq 100 then
    drop
  else
    if local-preference ge 101 then
      set med 87 ! prefix will have MED modified to 87
      pass       ! and passed
    else
      drop       ! this won't affect prefix processing, as it's only
                 ! checked if LP was anything else than "greater or equal to 101"
    endif
  endif
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
end-policy
!
```

Now, with LP value of 30:

```
RP/0/0/CPU0:SP1R2#show rpl route-policy IF-ELSE-IF
route-policy IF-ELSE-IF
  if local-preference eq 100 then
    drop
  else ! prefix doesn't match this branch
    if local-preference ge 101 then
      set med 87
      pass
    else ! prefix matches this value – it's not 100, it's not 101 or higher
      drop       ! ...so it will be dropped
    endif
  endif
end-policy
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 6.5: Nested RPL policies

RPL policies can be nested. Here's example of two child policies, each doing its own thing – TEST-CHILD-01 setting weight to 555, and TEST-CHILD-02 appending community value of 64999:666 to existing prefix communities.

They're applied together in TEST-PARENT policy that can be used to influence the way prefixes are received or advertised:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#route-policy TEST-CHILD-01
RP/0/0/CPU0:SP1R2(config-rpl)#set weight 555
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
RP/0/0/CPU0:SP1R2(config)#route-policy TEST-CHILD-02
RP/0/0/CPU0:SP1R2(config-rpl)#set community ( 64999:666 ) additive
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
RP/0/0/CPU0:SP1R2(config)#route-policy TEST-PARENT
RP/0/0/CPU0:SP1R2(config-rpl)#apply TEST-CHILD-01
RP/0/0/CPU0:SP1R2(config-rpl)#apply TEST-CHILD-02
RP/0/0/CPU0:SP1R2(config-rpl)#pass
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
RP/0/0/CPU0:SP1R2(config)#commit
RP/0/0/CPU0:SP1R2(config)#
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 6.6: Using RPL parameters

Sometimes it's useful to not define route-policy in static, but rather dynamic terms, like for example change specific prefix attribute but per neighbor. Here's such example – with local-preference being the dynamic parameter:

```
RP/0/0/CPU0:SP1R2(config)#route-policy SET-LP($LP)
RP/0/0/CPU0:SP1R2(config-rpl)#set local-preference $LP ! this will be dynamic attribute
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 1.1.1.1
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#remote-as 100
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy SET-LP (333) in ! set LP to 333
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#exit
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#exit
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 2.2.2.2
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#remote-as 200
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy SET-LP (777) in ! set LP to 777
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#commit          ! using the same route-policy
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 6.7: Viewing RPL status

To check which route-policies are defined you'd do:

```
RP/0/0/CPU0:SP1R2#sh rpl route-policy
Listing for all Route Policy objects

route-policy PASS
  pass
end-policy
!
route-policy SET-LP($LP)
  set local-preference $LP
end-policy
!
route-policy TEST01
  if destination-prefix in BAD-GUYS and community matches-any (64999:666) then
    set next-hop discard
  endif
[…]
```

However, to check state of each route-policy, there's another, very useful command:

```
RP/0/0/CPU0:SP1R2#sh rpl route-policy states

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

The following policies are (ACTIVE)
-----------------------------------------
route-policy PASS
  pass
end-policy
!
route-policy RED-BGP
[…]
The following policies are (INACTIVE)
-----------------------------------------
route-policy TEST-CHILD-01
  set weight 555
end-policy
[…]
The following policies are (UNUSED)
-----------------------------------------
route-policy CHECK-8
  if destination-prefix in BAD-GUYS then
[…]
```

Another, very useful tool is capability to check where specific route-policy is applied:

```
RP/0/0/CPU0:SP1R2#sh rpl route-policy PASS attachpoints

BGP Attachpoint: Neighbor

Neighbor/Group                          type  afi/safi   in/out  vrf name   bound by
-------------------------------------------------------------------------------------
40.60.27.7                              --    IPv4/uni   in   default    PASS
40.60.27.7                              --    IPv4/uni   out  default    PASS
2001:40:60:27::7                        --    IPv6/uni   in   default    PASS
2001:40:60:27::7                        --    IPv6/uni   out  default    PASS
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 7: Traffic filtering and manipulation

In some cases it's crucial to be able to filter out, or change direction of packet routing. Filtering can be done either by ACLs, or by using policy based routing constructs. Let's go through some options available.

Task 7.1: Access-lists and object-groups

Let's define a filter that will pass only traffic going to port 80 on one of our IPs while denying every other IPv4 traffic on both IOS-XE and IOS XR. On IOS-XE it's simple enough:

```
SP2-R5-RR#conf t
SP2-R5-RR(config)#ip access-list extended TEST_FILTER
SP2-R5-RR(config-ext-nacl)#permit tcp any host 12.39.5.5 eq 80
SP2-R5-RR(config-ext-nacl)#deny ip any any
SP2-R5-RR(config-ext-nacl)#exit

! and now let's apply the ACL for INGRESS traffic on Gi1:

SP2-R5-RR(config)#int gi1
SP2-R5-RR(config-if)#ip access-group TEST_FILTER in
```

On IOS XR, it's very similar, you can notice however a number of additional options while configuring:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#ipv4 access-list TEST_FILTER
RP/0/0/CPU0:SP2R2(config-ipv4-acl)#permit tcp any host 12.39.5.5 eq 80
RP/0/0/CPU0:SP2R2(config-ipv4-acl)#deny ipv4 any any
RP/0/0/CPU0:SP2R2(config-ipv4-acl)#commit

RP/0/0/CPU0:SP2R2#sh running-config ipv4 access-list
ipv4 access-list TEST_FILTER
 10 permit tcp any any eq www
 20 deny ipv4 any any

! and now let's apply the ACL for INGRESS traffic on GigabitEthernet0/0/0/1:

RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#interface gigabitEthernet 0/0/0/1
RP/0/0/CPU0:SP2R2(config-if)#ipv4 access-group TEST_FILTER ingress
RP/0/0/CPU0:SP2R2(config-if)#commit
RP/0/0/CPU0:SP2R2(config-if)#
RP/0/0/CPU0:SP2R2#
```

One of the options to verify ACL is applied (and in what direction) is to check the interface IPv4 or IPv6 status, like this (for IPv4):

```
RP/0/0/CPU0:SP2R2#sh ipv4 interface gigabitEthernet 0/0/0/1
GigabitEthernet0/0/0/1 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 20.40.27.2/24
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Multicast reserved groups joined: 224.0.0.2 224.0.0.1
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  common access list is not set, access list is TEST_FILTER
  Proxy ARP is disabled
  ICMP redirects are never sent
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 8: Basic IPv4 configuration

Interface configuration for IPv4 is exactly the same in IOS-XE as it was with IOS. Let's check the current configuration of Gi3 interface of CEL7:

```
CEL7#sh run int gi3
!
interface GigabitEthernet3
 description to CEL6
 ip address 44.44.67.7 255.255.255.0
 negotiation auto
 ipv6 address FE80::7 link-local
 ipv6 address 2001:44:44:67::7/64
 no mop enabled
 no mop sysid
end
```

As soon as you'll edit the config and define new IPv4 address for interface, the old one will be overwritten.

Also, in IOS-XE there's still option to use secondary addresses for IPv4:

```
CEL7#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
CEL7(config)#int gi3
CEL7(config-if)#ip address 45.45.45.1 255.255.255.0 secondary
CEL7#sh run int gi3
!
interface GigabitEthernet3
 description to CEL6
 ip address 45.45.45.1 255.255.255.0 secondary
 ip address 44.44.67.7 255.255.255.0
 negotiation auto
 ipv6 address FE80::7 link-local
 ipv6 address 2001:44:44:67::7/64
 no mop enabled
 no mop sysid
end
```

Please however note, that secondary IPv4 addresses have limitations as far as scope and functionality goes and are of limited use in real-world SP networks.

Please delete the secondary IPv4 address we've just created before moving forward.

Very similar configuration options are present with IOS XR, here however we need to use 'ipv4' prefix. There's however one bonus – you can use slash notation for subnet masks, like this:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#int gi0/0/0/1
RP/0/0/CPU0:SP1R2(config-if)#ipv4 address 40.60.27.2/24
RP/0/0/CPU0:SP1R2(config-if)#commit
RP/0/0/CPU0:SP1R2#sh run int gi0/0/0/1
!
interface GigabitEthernet0/0/0/1
 description to CEL7
 ipv4 address 40.60.27.2 255.255.255.0
 ipv6 address fe80::2 link-local
 ipv6 address 2001:40:60:27::2/64
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

IOS XR can also use ‘secondary’ IPv4 addressing, like it’s IOS and IOS-XE brothers:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#int gi0/0/0/1
RP/0/0/CPU0:SP1R2(config-if)#ipv4 address 45.45.45.1/24 secondary
RP/0/0/CPU0:SP1R2(config-if)#commit
RP/0/0/CPU0:SP1R2(config-if)#
RP/0/0/CPU0:SP1R2#sh run int gi0/0/0/1
interface GigabitEthernet0/0/0/1
 description to CEL7
 ipv4 address 40.60.27.2 255.255.255.0
 ipv4 address 45.45.45.1 255.255.255.0 secondary
 ipv6 address fe80::2 link-local
 ipv6 address 2001:40:60:27::2/64
!
```

Delete the secondary address before you continue to next task. Notice, with IOS XR it’s easy to do using rollback capability:

```
RP/0/0/CPU0:SP1R2#rollback configuration last 1
Loading Rollback Changes.
Loaded Rollback Changes in 1 sec
Committing.
2 items committed in 1 sec (1)items/sec
Updating.
Updated Commit database in 1 sec
Configuration successfully rolled back 1 commits.
RP/0/0/CPU0:SP1R2#sh run int gi0/0/0/1
interface GigabitEthernet0/0/0/1
 description to CEL7
 ipv4 address 40.60.27.2 255.255.255.0
 ipv6 address fe80::2 link-local
 ipv6 address 2001:40:60:27::2/64
!
```

The IPv4 secondary address is gone.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 9: Basic IPv6 configuration

With IPv6, both IOS-XE and IOS XR follow the notation of '**ipv6**' before specifying the address. Like this:

```
CEL7#conf t
CEL7(config)#int gi3
CEL7(config-if)#ipv6 address 2001:55:55:55::1/64
CEL7(config-if)#^Z
CEL7#sh run int gi3
interface GigabitEthernet3
 description to CEL6
 ip address 45.45.45.1 255.255.255.0 secondary
 ip address 44.44.67.7 255.255.255.0
 negotiation auto
 ipv6 address FE80::7 link-local
 ipv6 address 2001:44:44:67::7/64
 ipv6 address 2001:55:55:55::1/64
 no mop enabled
 no mop sysid
end
```

Notice however, that with IPv6 you can have many addresses assigned to interface – and they don't need to be '**secondary**'. That means, there's greater flexibility in terms of addressing the node, but unfortunately also – greater room for error and mistakes. While with IPv4 new address overwrites the old one, with IPv6 it's simply added.

The same holds true for IOS XR:

```
RP/0/0/CPU0:SP1R2#sh running-config interface gigabitEthernet 0/0/0/1
interface GigabitEthernet0/0/0/1
 description to CEL7
 ipv4 address 40.60.27.2 255.255.255.0
 ipv6 address fe80::2 link-local
 ipv6 address 2001:40:60:27::2/64
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 9.1: IPv6 address assignment in IOS-XE – EUI64

Apart from configuring the IPv6 address statically, you can use EUI-64 mechanism to translate MAC address of physical interface to automatically generated IPv6 address given prefix with /64 mask. On IOS-XE:

Let's check interface MAC:

```
CEL6#sh int gi2 | i bia
  Hardware is CSR vNIC, address is fa16.3efa.1c6f (bia fa16.3efa.1c6f)
```

Interface config shouldn't have any IPv6 defined (watch out for link-local definition – if there's one, EUI-64 will be essentially static):

```
CEL6#sh run int gi2
interface GigabitEthernet2
 description to SP1R1
 ip address 41.61.16.6 255.255.255.0
 negotiation auto
 no mop enabled
 no mop sysid
end
```

Now, let's configure IPv6 EUI-64 address:

```
CEL6#conf t
CEL6(config-if)#ipv6 address 2001:41:61:17::/64 eui-64
CEL6(config-if)#^Z
```

And now we have a prefix:

```
CEL6#sh ipv6 interface gigabitEthernet 2
GigabitEthernet2 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::F816:3EFF:FEFA:1C6F
  No Virtual link-local address(es):
  Description: to SP1R1
  Global unicast address(es):
    2001:41:61:16:F816:3EFF:FEFA:1C6F, subnet is 2001:41:61:16::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FFFA:1C6F
  MTU is 1500 bytes
```

…on IOS-XE, and similar command on IOS XR:

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 9.2: IPv6 address assignment in IOS XR – EUI64

We can repeat the same exercise with IOS XR:

```
RP/0/0/CPU0:SP1R2#sh int gi0/0/0/3 | i bia
  Hardware is GigabitEthernet, address is fa16.3e53.aab4 (bia fa16.3e53.aab4)
```

The interface itself is clean out of IPv6 configuration:

```
RP/0/0/CPU0:SP1R2#sh run int gi0/0/0/3
interface GigabitEthernet0/0/0/3
 description to CEL6
 ipv4 address 41.61.26.2 255.255.255.0
!
```

Now, let's config EUI-64 address:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#int gi0/0/0/3
RP/0/0/CPU0:SP1R2(config-if)#ipv6 address 2001:41:61:26::/64 eui-64
RP/0/0/CPU0:SP1R2(config-if)#commit
RP/0/0/CPU0:SP1R2(config-if)#
```

And we should have it:

```
RP/0/0/CPU0:SP1R2#sh ipv6 int gi0/0/0/3
GigabitEthernet0/0/0/3 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::f816:3eff:fe53:aab4
  Global unicast address(es):
    2001:41:61:26:f816:3eff:fe53:aab4, subnet is 2001:41:61:26::/64
  Joined group address(es): ff02::1:ff53:aab4 ff02::2 ff02::1
  MTU is 1514 (1500 is available to IPv6)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 9.3: IPv6 neighbors

When IPv6 is activated on interface, NDP (Neighbor Discovery Protocol) uses it's communication with ICMPv6 to detect neighbors over the link. This can be verified with:

```
CEL7#sh ipv6 neighbors
IPv6 Address                              Age Link-layer Addr State Interface
FE80::2                                     2 fa16.3e91.bc00  STALE Gi2
```

…on IOS-XE, and similar command on IOS XR:

```
RP/0/0/CPU0:SP1R2#sh ipv6 neighbors
IPv6 Address                 Age Link-layer Addr State Interface            Location
fe80::3                    29854 fa16.3ecf.1bfd GLEAN Gi0/0/0/0             0/0/CPU0
[Mcast adjacency]              - 0000.0000.0000 REACH Gi0/0/0/0             0/0/CPU0
fe80::7                    28021 fa16.3e3d.674c DELAY Gi0/0/0/1             0/0/CPU0
[Mcast adjacency]              - 0000.0000.0000 REACH Gi0/0/0/1             0/0/CPU0
fe80::1                    29886 fa16.3e85.e41f GLEAN Gi0/0/0/2             0/0/CPU0
[Mcast adjacency]              - 0000.0000.0000 REACH Gi0/0/0/2             0/0/CPU0
fe80::6                    29850 fa16.3e9f.657b GLEAN Gi0/0/0/3             0/0/CPU0
[Mcast adjacency]              - 0000.0000.0000 REACH Gi0/0/0/3             0/0/CPU0
fe80::4                    29856 fa16.3e42.ef88 GLEAN Gi0/0/0/4             0/0/CPU0
[Mcast adjacency]              - 0000.0000.0000 REACH Gi0/0/0/4             0/0/CPU0
```

Notice, that we selected to use link-local addresses matching the router number on all interfaces. This disables self-generated link-local addresses, that would make troubleshooting harder.

Let's delete the defined link-local addresses and check the generated one:

```
CEL7#sh ipv6 interface gigabitEthernet 3
GigabitEthernet3 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::7
  No Virtual link-local address(es):
  Description: to CEL6
  Global unicast address(es):
    2001:44:44:67::7, subnet is 2001:44:44:67::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:7
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.
CEL7#conf t
CEL7(config)#int gi3
CEL7(config-if)#no ipv6 address FE80::7 link-local
CEL7(config-if)#^Z
CEL7#sh ipv6 interface gigabitEthernet 3
GigabitEthernet3 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::F816:3EFF:FEAC:8A00
  No Virtual link-local address(es):
  Description: to CEL6
  Global unicast address(es):
    2001:44:44:67::7, subnet is 2001:44:44:67::/64 [TEN]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:7
    FF02::1:FFAC:8A00
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
Hosts use stateless autoconfig for addresses.
```

Making the link-local addresses stick to some predefined format helps troubleshooting. The EUI64 format that it's used for link-local addresses when there's no explicit configuration makes it dependent on the physical interface MAC address. In service provider networks, changing line cards with interfaces is not that rare, and changing the addressing can be problematic. In environments when this address is important (like local access Ethernet segments) defining it upfront is best practice.

Let's configure the link-local address on Gi3 of CEL7 again:

```
CEL7#conf t
CEL7(config)#int gi3
CEL7(config-if)#ipv6 address FE80::7 link-local
CEL7(config-if)#^Z
```

Now, let's ping the other side of the link – SP1R2 router using the link-local address of fe80::2 (2 as R2):

```
CEL7#ping fe80::2
Output Interface: GigabitEthernet2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::2, timeout is 2 seconds:
Packet sent with a source address of FE80::7%GigabitEthernet2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/18 ms
```

Note, you need to specify the output interface – as link-local addresses are just that – specific to a link.

You can shortcut the ping using '**%**' sign and interface specification, like that:

```
CEL7#ping fe80::2%GigabitEthernet2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::2, timeout is 2 seconds:
Packet sent with a source address of FE80::7%GigabitEthernet2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

The same trick doesn't unfortunately work yet on IOS XR:

```
RP/0/0/CPU0:SP1R2#ping fe80::7
Output Interface: gi0/0/0/1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to fe80::7, timeout is 2 seconds:
!!!!!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/9 ms
RP/0/0/CPU0:SP1R2#ping fe80::7%gi0/0/0/1
% Bad hostname or protocol not running
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 9.4: IPv6 SLAAC

Let's configure SLAAC, with IOS XR serving IPv6 addresses and IOS-XE receiving them from a pool.

We'll use SP1R2 and CEL7 for the exercise.

On SP1R2:

```
RP/0/0/CPU0:SP1R2(config)#int gi0/0/0/1
RP/0/0/CPU0:SP1R2(config-if)#ipv6 address 2006:10::1/64
RP/0/0/CPU0:SP1R2(config-if)#ipv6 nd prefix 2006:10::/64 infinite infinite
RP/0/0/CPU0:SP1R2(config-if)#ipv6 nd ra-interval 10
RP/0/0/CPU0:SP1R2(config-if)#commit
RP/0/0/CPU0:SP1R2(config-if)#
```

On CEL7:

```
CEL7#conf t
CEL7(config)#int gi2
CEL7(config-if)#ipv6 address autoconfig
CEL7(config-if)#^Z
CEL7#
```

CEL7 should get IPv6 from the pool specified on SP1R2:

```
CEL7#sh ipv6 interface gigabitEthernet 2
GigabitEthernet2 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::7
  No Virtual link-local address(es):
  Description: to SP1R2
  Stateless address autoconfig enabled
  Global unicast address(es):
    2006:10::7, subnet is 2006:10::/64 [EUI/CAL/PRE]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:7
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 9.5: IPv6 DHCP

CEL6 will assign addresses using IPv6 DHCP and CEL7 will receive them.

First, let's configure to interface upstream to CEL7 on CEL6:

```
interface GigabitEthernet3
 ipv6 address FE80::6 link-local
 ipv6 address 2016:100::1/64
 ipv6 nd other-config-flag
```

Now, let's configure the IPv6 pool:

```
ipv6 dhcp pool IPV6_POOL
 prefix-delegation pool SPBCPL_POOL lifetime 600 600
 dns-server 2001:DB8::1
 domain-name spbcpl.com
!
ipv6 local pool SPBCPL_POOL 2016:100::/48 64
```

And now, attach it to the interface:

```
interface GigabitEthernet3
 ipv6 dhcp server IPV6_POOL
```

CEL7 now requires reconfiguration for properly receiving the prefix – RA will signal that the information should be obtained via 'other' information source – DHCPv6:

```
CEL7#sh run int gi3
interface GigabitEthernet3
 description to CEL6
 ip address 44.44.67.7 255.255.255.0
 negotiation auto
 ipv6 address autoconfig
 ipv6 address FE80::7 link-local
 no mop enabled
 no mop sysid
end
```

CEL7 should get the prefix:

```
CEL7#sh ipv6 int gi3
GigabitEthernet3 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::7
  No Virtual link-local address(es):
  Description: to CEL6
  Stateless address autoconfig enabled
  Global unicast address(es):
    2016:100::7, subnet is 2016:100::/64 [EUI/CAL/PRE]
      valid lifetime 2591991 preferred lifetime 604791
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:7
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  [...]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

So let's ping the "default gateway" – CEL6 from CEL7:

```
CEL7#ping 2016:100::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2016:100::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/17 ms
```

You can also verify the DHCPv6 server status on CEL6:

```
CEL6#sh ipv6 dhcp pool
DHCPv6 pool: IPV6_POOL
  Prefix pool: SPBCPL_POOL
               preferred lifetime 600, valid lifetime 600
  DNS server: 2001:DB8::1
  Domain name: spbcpl.com
  Active clients: 0
CEL6#sh ipv6 dhcp statistics
Messages received                9
Messages sent                    9
Messages discarded               0
Messages could not be sent       0

Messages                         Received
SOLICIT                          8
INFORMATION-REQUEST              1

Messages                         Sent
ADVERTISE                        8
REPLY                            1
```

And verify the interface with the IPv6 server attached:

```
CEL6#sh ipv6 dhcp interface
GigabitEthernet3 is in server mode
  Using pool: IPV6_POOL
  Preference value: 0
  Hint from client: ignored
  Rapid-Commit: disabled
```



Before moving forward after playing with various options of IOS-XE and IOS XR, please stop and restart your pod with clean configuration for IPv4+IPv6. Otherwise, some features may fail to work with interfaces configured or reconfigured in a way that breaks connectivity.

The proper starting file is **SP CCIE bootcamp clean config - week 1.unl**.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 3: OSPF & ISIS

Tricky OSPF

In this lab you will be presented examples of tricky configuration in OSPF which may cause issues in the network. Note that some Router-IDs can be different than in your environment which does not change configuration steps and results unless stated otherwise.

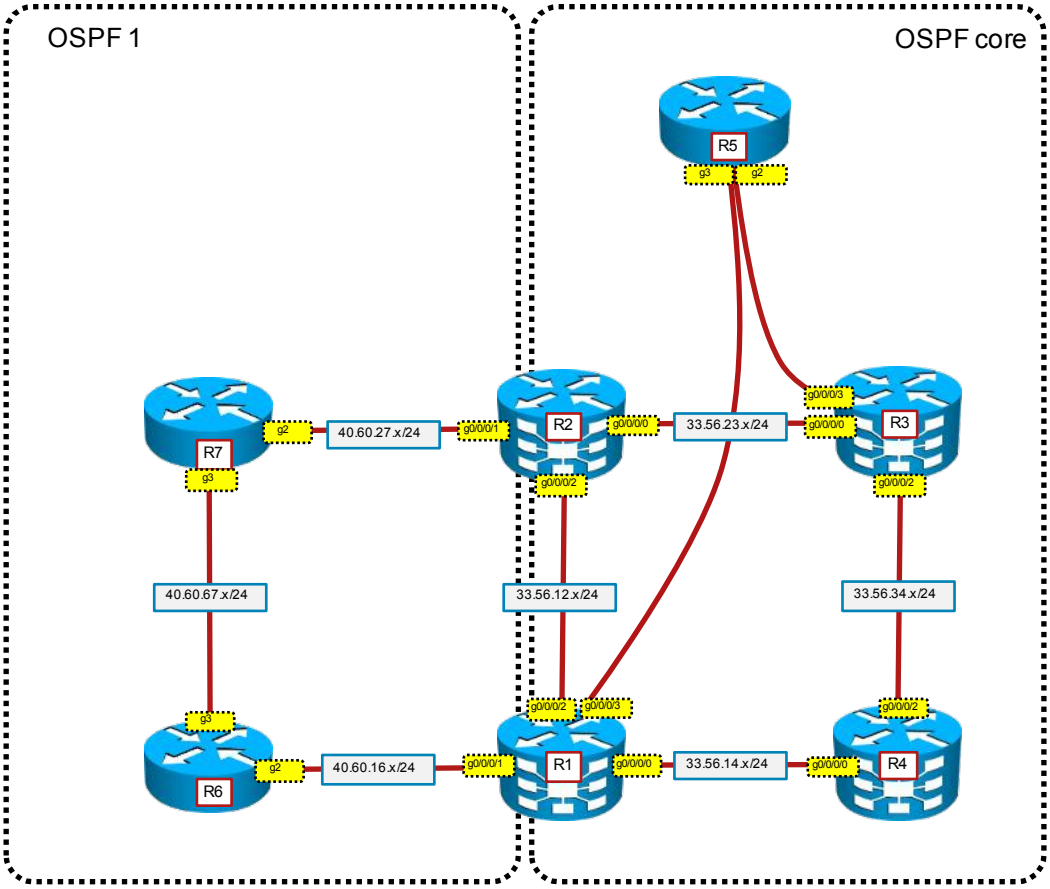
This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch03.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch03.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch03.unl`

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Topology 1: AS3356 OSPF

There are two OSPF processes in AS 3356. There is a mutual redistribution of prefixed between two OSPF domains.



There are intra-, inter-area and external prefixes learnt through OSPF.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The routing table from SP1R2:

```
RP/0/0/CPU0:SP1R2#sh route
Mon Jan  1 21:39:35.211 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

Gateway of last resort is not set

 O E2 11.11.11.0/24 [110/20] via 33.56.23.3, 00:00:06, GigabitEthernet0/0/0/0
 O E2 33.56.1.1/32 [110/1] via 40.60.27.7, 00:00:33, GigabitEthernet0/0/0/1
 L    33.56.2.2/32 is directly connected, 00:00:37, Loopback0
 O IA 33.56.3.3/32 [110/101] via 33.56.23.3, 00:00:31, GigabitEthernet0/0/0/0
 O IA 33.56.4.4/32 [110/102] via 33.56.23.3, 00:00:00, GigabitEthernet0/0/0/0
 O E2 33.56.5.5/32 [110/101] via 40.60.27.7, 00:00:33, GigabitEthernet0/0/0/1
 C    33.56.12.0/24 is directly connected, 00:00:37, GigabitEthernet0/0/0/2
 L    33.56.12.2/32 is directly connected, 00:00:37, GigabitEthernet0/0/0/2
 O E2 33.56.14.0/24 [110/20] via 40.60.27.7, 00:00:33, GigabitEthernet0/0/0/1
 O E2 33.56.15.0/24 [110/20] via 40.60.27.7, 00:00:33, GigabitEthernet0/0/0/1
 C    33.56.23.0/24 is directly connected, 00:00:37, GigabitEthernet0/0/0/0
 L    33.56.23.2/32 is directly connected, 00:00:37, GigabitEthernet0/0/0/0
 C    33.56.24.0/24 is directly connected, 00:00:37, GigabitEthernet0/0/0/4
 L    33.56.24.2/32 is directly connected, 00:00:37, GigabitEthernet0/0/0/4
 O IA 33.56.34.0/24 [110/101] via 33.56.23.3, 00:00:31, GigabitEthernet0/0/0/0
 O IA 33.56.35.0/24 [110/101] via 33.56.23.3, 00:00:31, GigabitEthernet0/0/0/0
 O    40.60.16.0/24 [110/3] via 40.60.27.7, 00:00:33, GigabitEthernet0/0/0/1
 O    40.60.17.0/24 [110/2] via 40.60.27.7, 00:00:33, GigabitEthernet0/0/0/1
 C    40.60.27.0/24 is directly connected, 00:00:37, GigabitEthernet0/0/0/1
 L    40.60.27.2/32 is directly connected, 00:00:37, GigabitEthernet0/0/0/1
 C    41.61.26.0/24 is directly connected, 00:00:37, GigabitEthernet0/0/0/3
 L    41.61.26.2/32 is directly connected, 00:00:37, GigabitEthernet0/0/0/3
 O E2 41.61.46.0/24 [110/20] via 33.56.23.3, 00:00:06, GigabitEthernet0/0/0/0
 O    44.44.67.0/24 [110/2] via 40.60.27.7, 00:00:33, GigabitEthernet0/0/0/1
 O E2 172.16.6.6/32 [110/2] via 33.56.23.3, 00:00:31, GigabitEthernet0/0/0/0
 O    172.16.7.7/32 [110/2] via 40.60.27.7, 00:00:33, GigabitEthernet0/0/0/1
 O    172.16.77.77/32 [110/2] via 40.60.27.7, 00:00:33, GigabitEthernet0/0/0/1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: Loopback interface prefix

Cisco IOS, IOS-XE and IOS XR advertises prefixes linked with Loopback interfaces as host routes. The advertised mask in OSPF is critical and must match the mask configured on the interface. Let's check the current settings and change them.

The route 33.56.4.4/32 is advertised by SP1R4.

```
RP/0/0/CPU0:SP1R3#sh route 33.56.4.4
Sun Jan  1 00:40:58.717 UTC

Routing entry for 33.56.4.4/32
  Known via "ospf core", distance 110, metric 2, type intra area
  Installed Jan  1 22:35:28.963 for 02:05:29
  Routing Descriptor Blocks
    33.56.34.4, from 33.56.4.4, via GigabitEthernet0/0/0/2
      Route metric is 2
  No advertising protos.
```

Locate the source of the IP address.

```
RP/0/0/CPU0:SP1R4#sh ip int brief | i 33.56.4.4
Sun Jan  1 00:44:16.283 UTC
Loopback0                      33.56.4.4       Up              Up       default

RP/0/0/CPU0:SP1R4#sh int lo0
Sun Jan  1 00:45:27.598 UTC
Loopback0 is up, line protocol is up
  Interface state transitions: 3
  Hardware is Loopback interface(s)
  Description: Loopback
  Internet address is 33.56.4.4/32

RP/0/0/CPU0:SP1R4#sh run int lo0
Sun Jan  1 00:46:30.014 UTC
interface Loopback0
 description Loopback
 ipv4 address 33.56.4.4 255.255.255.255
!
```

Change the mask of the interface to /24.

```
RP/0/0/CPU0:SP1R4(config)#int lo0
RP/0/0/CPU0:SP1R4(config-if)#ipv4 add 33.56.4.4/24
RP/0/0/CPU0:SP1R4(config-if)#commit
```

Now, there are two prefixes on SP1R2 33.56.4.4/32 and 33.56.4.0/24. Why?

```
RP/0/0/CPU0:SP1R2#sh route longer-prefixes 33.56.4.0/24
Mon Jan  1 22:15:49.292 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
[..]
O E2 33.56.4.0/24 [110/20] via 33.56.12.1, 00:07:36, GigabitEthernet0/0/0/2
O IA 33.56.4.4/32 [110/102] via 33.56.23.3, 19:25:45, GigabitEthernet0/0/0/0
                  [110/102] via 33.56.12.1, 19:25:45, GigabitEthernet0/0/0/2
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

One prefix is inter-area, an another one is type external tagged with 44.

```
RP/0/0/CPU0:SP1R2#sh route 33.56.4.4

Routing entry for 33.56.4.4/32
  Known via "ospf core", distance 110, metric 102, type inter area
  Installed Jan  1 02:50:04.014 for 19:29:14
  Routing Descriptor Blocks
    33.56.23.3, from 33.56.3.3, via GigabitEthernet0/0/0/0
      Route metric is 102
    33.56.12.1, from 33.56.4.4, via GigabitEthernet0/0/0/2
      Route metric is 102
  No advertising protos.

RP/0/0/CPU0:SP1R2#sh route 33.56.4.0

Routing entry for 33.56.4.0/24
  Known via "ospf core", distance 110, metric 20
  Tag 44, type extern 2
  Installed Jan  1 22:08:12.783 for 00:11:26
  Routing Descriptor Blocks
    33.56.12.1, from 33.56.4.4, via GigabitEthernet0/0/0/2
      Route metric is 20
  No advertising protos.
```

For the prefix 33.56.4.4/32 the source of information within the area is ABR R3 and SP1R4. For the prefix 33.56.4.0/24 the source of information within the area is ASBR which is SP1R4.

```
RP/0/0/CPU0:SP1R2#sh ospf core database external 33.56.4.0

            OSPF Router with ID (33.56.23.2) (Process ID core)
                Type-5 AS External Link States

  Routing Bit Set on this LSA
  LS age: 1242
  Options: (No TOS-capability, DC)
  LS Type: AS External Link
  Link State ID: 33.56.4.0 (External Network Number)
  Advertising Router: 33.56.4.4
  LS Seq Number: 80000001
  Checksum: 0x6550
  Length: 36
  Network Mask: /24
        Metric Type: 2 (Larger than any link state path)
        TOS: 0
        Metric: 20
        Forward Address: 0.0.0.0
        External Route Tag: 44
```

Check the configuration on R4. There are two possible sources of the prefix; through the redistribution and the interface's prefix advertisement.

```
RP/0/0/CPU0:SP1R4#sh protocols ospf

Routing Protocol OSPF core
  Router Id: 33.56.4.4
  Distance: 110
  Non-Stop Forwarding: Disabled
  Redistribution:
    connected
  Area 0
    Loopback0
    GigabitEthernet0/0/0/2
  Area 1
    GigabitEthernet0/0/0/0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Find the mask of the external route.

```
RP/0/0/CPU0:SP1R4#sh ospf data external 33.56.4.0

            OSPF Router with ID (33.56.4.4) (Process ID core)

                Type-5 AS External Link States

  LS age: 1888
  Options: (No TOS-capability, DC)
  LS Type: AS External Link
  Link State ID: 33.56.4.0 (External Network Number)
  Advertising Router: 33.56.4.4
  LS Seq Number: 80000001
  Checksum: 0x6550
  Length: 36
  Network Mask: /24
        Metric Type: 2 (Larger than any link state path)
        TOS: 0
        Metric: 20
        Forward Address: 0.0.0.0
        External Route Tag: 44
```

Find the mask of the internal route. We know from the previous output that Loopback 0 is in area 0.

```
RP/0/0/CPU0:SP1R4#sh ospf core 0 database router 33.56.4.4 adv-router 33.56.4.4

            OSPF Router with ID (33.56.4.4) (Process ID core)

                Router Link States (Area 0)

  LS age: 1054
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 33.56.4.4
  Advertising Router: 33.56.4.4
  LS Seq Number: 8000002d
  Checksum: 0x4a85
  Length: 48
  Area Border Router
  AS Boundary Router
   Number of Links: 2

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 33.56.4.4
     (Link Data) Network Mask: 255.255.255.255
      Number of TOS metrics: 0
       TOS 0 Metrics: 1
[..]
```

As you can see the network mask did not change the advertised mask which is still /32. Here is how to change it. You need to define loopback type as stub.

```
RP/0/0/CPU0:SP1R4(config)#router ospf core
RP/0/0/CPU0:SP1R4(config-ospf)#area 0
RP/0/0/CPU0:SP1R4(config-ospf-ar)#int loopback 0
RP/0/0/CPU0:SP1R4(config-ospf-ar-if)#loopback stub-network enable
RP/0/0/CPU0:SP1R4(config-if)#commit
```

To advertise the loopbacks configured mask IOS and IOS XE you need to define loopbacks network type as point-to-point.

```
SP1-R5-RR#sh ip ospf int brief
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Lo0          1     0               33.56.5.5/32        1     LOOP  0/0
Gi3          1     0               33.56.15.5/24       1     DR    1/1
Gi2          1     0               33.56.35.5/24       1     DR    1/1

SP1-R5-RR(config)#int lo0
SP1-R5-RR(config-if)#ip ospf network point-to-point

SP1-R5-RR#sh ip ospf int brief
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Lo0          1     0               33.56.5.5/32        1     P2P   0/0
Gi3          1     0               33.56.15.5/24       1     DR    1/1
Gi2          1     0               33.56.35.5/24       1     DR    1/1
```

Verification of the correct /24 mask on XR after changing the type of loopback to stub.

```
RP/0/0/CPU0:SP1R4#sh ospf core 0 database router 33.56.4.4 adv-router 33.56.4.4

            OSPF Router with ID (33.56.4.4) (Process ID core)

                Router Link States (Area 0)

  LS age: 23
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 33.56.4.4
  Advertising Router: 33.56.4.4
  LS Seq Number: 8000002e
  Checksum: 0x20b2
  Length: 48
  Area Border Router
  AS Boundary Router
   Number of Links: 2

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 33.56.4.0
     (Link Data) Network Mask: 255.255.255.0
      Number of TOS metrics: 0
       TOS 0 Metrics: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 33.56.34.4
     (Link Data) Router Interface address: 33.56.34.4
      Number of TOS metrics: 0
       TOS 0 Metrics: 1
```

On R2 there is 33.56.4.0/24. But why there is still the prefix 33.56.4.4/32 advertised as external E2?

```
RP/0/0/CPU0:SP1R2#sh route longer-prefixes 33.56.4.0/24

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

O IA 33.56.4.0/24 [110/102] via 33.56.23.3, 00:02:14, GigabitEthernet0/0/0/0
                  [110/102] via 33.56.12.1, 00:02:14, GigabitEthernet0/0/0/2
O E2 33.56.4.4/32 [110/20] via 33.56.12.1, 00:02:14, GigabitEthernet0/0/0/2
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Why do prefixes persist?

Before changing the R4's loopback IP address mask from /32 to /24 there was just one prefix in the routing table on R2.

```
O IA 33.56.4.4/32 [110/102] via 33.56.23.3, 18:49:31, GigabitEthernet0/0/0/0
                  [110/102] via 33.56.12.1, 18:49:31, GigabitEthernet0/0/0/2
```

After the change to /24 there were two prefixes /24 and /32.

```
O E2 33.56.4.0/24 [110/20] via 33.56.12.1, 00:07:36, GigabitEthernet0/0/0/2
O IA 33.56.4.4/32 [110/102] via 33.56.23.3, 19:25:45, GigabitEthernet0/0/0/0
                  [110/102] via 33.56.12.1, 19:25:45, GigabitEthernet0/0/0/2
```

But after making a correction on the loopback type we should have just /24.

There are still two prefixes. Why?

```
O IA 33.56.4.0/24 [110/102] via 33.56.23.3, 00:02:14, GigabitEthernet0/0/0/0
                  [110/102] via 33.56.12.1, 00:02:14, GigabitEthernet0/0/0/2
O E2 33.56.4.4/32 [110/20] via 33.56.12.1, 00:02:14, GigabitEthernet0/0/0/2
```

Check who is the source of the /32 prefix.

```
RP/0/0/CPU0:SP1R2#sh ospf database external 33.56.4.4
Tue Jan  1 00:19:23.894 UTC


            OSPF Router with ID (33.56.2.2) (Process ID 1)

                Type-5 AS External Link States

  LS age: 532
  Options: (No TOS-capability, DC)
  LS Type: AS External Link
  Link State ID: 33.56.4.4 (External Network Number)
  Advertising Router: 33.56.2.2
  LS Seq Number: 80000001
  Checksum: 0x731c
  Length: 36
  Network Mask: /32
        Metric Type: 2 (Larger than any link state path)
        TOS: 0
        Metric: 102
        Forward Address: 0.0.0.0
        External Route Tag: 0



            OSPF Router with ID (33.56.23.2) (Process ID core)
  Routing Bit Set on this LSA
  LS age: 533
  Options: (No TOS-capability, DC)
  LS Type: AS External Link
  Link State ID: 33.56.4.4 (External Network Number)
  Advertising Router: 33.56.14.1
  LS Seq Number: 80000001
  Checksum: 0x255f
  Length: 36
  Network Mask: /32
        Metric Type: 2 (Larger than any link state path)
        TOS: 0
        Metric: 102
        Forward Address: 0.0.0.0
        External Route Tag: 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R1#sh ospf data external 33.56.4.4
Tue Jan  1 00:23:13.078 UTC


            OSPF Router with ID (33.56.1.1) (Process ID 1)

                Type-5 AS External Link States

  Routing Bit Set on this LSA
  LS age: 764
  Options: (No TOS-capability, DC)
  LS Type: AS External Link
  Link State ID: 33.56.4.4 (External Network Number)
  Advertising Router: 33.56.2.2
  LS Seq Number: 80000001
  Checksum: 0x731c
  Length: 36
  Network Mask: /32
        Metric Type: 2 (Larger than any link state path)
        TOS: 0
        Metric: 102
        Forward Address: 0.0.0.0
        External Route Tag: 0



            OSPF Router with ID (33.56.14.1) (Process ID core)
  LS age: 761
  Options: (No TOS-capability, DC)
  LS Type: AS External Link
  Link State ID: 33.56.4.4 (External Network Number)
  Advertising Router: 33.56.14.1
  LS Seq Number: 80000001
  Checksum: 0x255f
  Length: 36
  Network Mask: /32
        Metric Type: 2 (Larger than any link state path)
        TOS: 0
        Metric: 102
        Forward Address: 0.0.0.0
        External Route Tag: 0
```

So R2 is pointing to R1 through the OSPF process core but R1 is pointing to R2 through the OSPF process 1.

Shut the loopback0 interface on R4 to see what is a change.

```
RP/0/0/CPU0:SP1R4(config)#int lo0
RP/0/0/CPU0:SP1R4(config-if)#shut
RP/0/0/CPU0:SP1R4(config-if)#commit
```

There is a potential loop here. Let's verify.

```
RP/0/0/CPU0:SP1R2#traceroute 33.56.4.4
Tue Jan  1 00:31:32.464 UTC

Type escape sequence to abort.
Tracing the route to 33.56.4.4

 1  40.60.27.7 0 msec  0 msec  0 msec
 2  40.60.67.6 0 msec  0 msec  0 msec
 3  40.60.16.1 0 msec  0 msec  0 msec
 4  33.56.12.2 0 msec  0 msec  0 msec
 5  40.60.27.7 0 msec  0 msec  0 msec
 6  40.60.67.6 0 msec  0 msec  0 msec
 7  40.60.16.1 0 msec  0 msec  0 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
 8  33.56.12.2 0 msec  9 msec  0 msec
 9  40.60.27.7 0 msec  0 msec  0 msec
 10 40.60.67.6 0 msec  0 msec  0 msec
 11 40.60.16.1 0 msec  0 msec  0 msec
 12 33.56.12.2 0 msec  0 msec  0 msec
 13 40.60.27.7 0 msec  0 msec  0 msec
 14 40.60.67.6 0 msec  0 msec  0 msec
 15 40.60.16.1 0 msec  0 msec  0 msec
 16 33.56.12.2 9 msec  0 msec  0 msec
 17 40.60.27.7 0 msec  0 msec  0 msec
 18 40.60.67.6 9 msec  9 msec  0 msec
 19 40.60.16.1 9 msec  9 msec  9 msec
 20 33.56.12.2 9 msec  0 msec  0 msec
 21 40.60.27.7 9 msec  9 msec  9 msec
 22 40.60.67.6 9 msec  9 msec  9 msec
 23 40.60.16.1 9 msec  9 msec  9 msec
 24 33.56.12.2 9 msec  9 msec  9 msec
 25 40.60.27.7 9 msec  9 msec  9 msec
 26 40.60.67.6 9 msec  9 msec  9 msec
 27 40.60.16.1 9 msec  9 msec  9 msec
 28 33.56.12.2 9 msec  9 msec  9 msec
 29 40.60.27.7 9 msec  9 msec  9 msec
 30 40.60.67.6 9 msec  9 msec  9 msec
```

A loop is there! Anticlockwise.

If you do 'no shut' there is a loop anyway.

```
RP/0/0/CPU0:SP1R2#traceroute 33.56.4.4
Tue Jan  1 00:31:09.716 UTC

Type escape sequence to abort.
Tracing the route to 33.56.4.4

 1  40.60.27.7 0 msec  0 msec  0 msec
 2  44.44.67.6 0 msec  0 msec  0 msec
 3  40.60.16.1 0 msec  0 msec  0 msec
 4  33.56.12.2 0 msec  0 msec  0 msec
 5  40.60.27.7 0 msec  0 msec  0 msec
 6  44.44.67.6 0 msec  0 msec  0 msec
 7  40.60.16.1 0 msec  0 msec  0 msec
 8  33.56.12.2 0 msec  0 msec  0 msec
 9  40.60.27.7 0 msec  0 msec  9 msec
 10 44.44.67.6 0 msec  0 msec  0 msec
 11 40.60.16.1 0 msec  0 msec  0 msec
 12 33.56.12.2 0 msec  0 msec  0 msec
 13 40.60.27.7 0 msec  0 msec  0 msec
 14 44.44.67.6 0 msec  0 msec  0 msec
 15 40.60.16.1 9 msec  9 msec  9 msec
 16 33.56.12.2 0 msec  9 msec  0 msec
 17 40.60.27.7 9 msec  9 msec  9 msec
 18 44.44.67.6 9 msec  9 msec  9 msec
 19 40.60.16.1 9 msec  9 msec  9 msec
 20 33.56.12.2 9 msec  9 msec  9 msec
 21 40.60.27.7 9 msec  9 msec  9 msec
 22 44.44.67.6 9 msec  9 msec  9 msec
 23 40.60.16.1 9 msec  9 msec  9 msec
 24 33.56.12.2 9 msec  9 msec  9 msec
 25 40.60.27.7 9 msec  9 msec  9 msec
 26 44.44.67.6 9 msec  9 msec  9 msec
 27 40.60.16.1 9 msec  9 msec  9 msec
 28 33.56.12.2 9 msec  9 msec  9 msec
 29 40.60.27.7 19 msec  19 msec  19 msec
 30 44.44.67.6 19 msec  19 msec  19 msec
```

Add a new loopback interface with the IP address.

```
RP/0/0/CPU0:SP1R4(config)#int lo1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R4(config-if)#ip add 33.56.44.44/32
RP/0/0/CPU0:SP1R4(config)#router ospf core
RP/0/0/CPU0:SP1R4(config-ospf)#area 0
RP/0/0/CPU0:SP1R4(config-ospf-ar)#interface lo1
RP/0/0/CPU0:SP1R4(config)#commit
```

Verify the prefix.

```
RP/0/0/CPU0:SP1R2#sh route longer-prefixes 33.56.44.0/24
Tue Jan  1 00:50:53.015 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

O IA 33.56.44.44/32 [110/102] via 33.56.23.3, 00:00:45, GigabitEthernet0/0/0/0
                    [110/102] via 33.56.12.1, 00:00:45, GigabitEthernet0/0/0/2
```

Check if there is a loop with traceroute to 33.56.44.44.

Now shut the loopback 0 and 1 on R4 so both loopbacks are shutdown.

```
RP/0/0/CPU0:SP1R4#sh ip int brief
Tue Jan  1 00:36:59.917 UTC

Interface                      IP-Address      Status          Protocol Vrf-Name
Loopback0                      33.56.4.4       Shutdown        Down     default
Loopback1                      33.56.44.44     Shutdown        Down     default

RP/0/0/CPU0:SP1R2#sh route longer-prefixes 33.56.44.0/24
Tue Jan  1 00:52:30.678 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

O E2 33.56.44.44/32 [110/102] via 33.56.12.1, 00:00:03, GigabitEthernet0/0/0/2
```

Check if there is a loop with traceroute to 33.56.44.44 even if there is no active interface for this IP address. You can add more interfaces to see that their prefixes persist in this network.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3: Why there is a loop?

Enable loopback interfaces on R4 by 'rollback configuration last 1'. You should see two equal cost paths to 33.56.4.0/24.

```
RP/0/0/CPU0:SP1R2#sh route 33.56.4.0
Tue Jan  1 02:21:12.124 UTC

Routing entry for 33.56.4.0/24
  Known via "ospf core", distance 110, metric 102, type inter area
  Installed Jan  1 02:20:14.218 for 00:00:57
  Routing Descriptor Blocks
    33.56.23.3, from 33.56.3.3, via GigabitEthernet0/0/0/0
      Route metric is 102
    33.56.12.1, from 33.56.4.4, via GigabitEthernet0/0/0/2
      Route metric is 102
  No advertising protos.
```

There is a loop for the path via 33.56.12.1.

```
RP/0/0/CPU0:SP1R2#trace 33.56.4.4

Type escape sequence to abort.
Tracing the route to 33.56.4.4

 1  33.56.12.1 0 msec  0 msec  0 msec
 2  40.60.16.6 0 msec  0 msec  0 msec
 3  40.60.67.7 0 msec  0 msec  0 msec
 4  40.60.27.2 0 msec  0 msec  0 msec
 5  33.56.12.1 0 msec  0 msec  0 msec
      [omitted]
```

There is also a different path via 33.56.23.3. Can you reach a destination IP? Yes, you need to find a different hash result to use an another path.

```
RP/0/0/CPU0:SP1R2#trace 33.56.4.3

Type escape sequence to abort.
Tracing the route to 33.56.4.4

 1  33.56.23.3 9 msec  0 msec  0 msec
 2  33.56.34.4 0 msec  *  0 msec
```

How to avoid a loop here for the destination 33.56.4.4? The issue is in the mutual redistribution between OSPF processes. Prefixes which are redistributed must not be redistributed back again. Apply the following configuration on R1 and R2.

```
!
route-policy 44-out
  if tag eq 44 then
    drop
  else
    pass
  endif
end-policy
!
!
router ospf 1
 redistribute ospf core tag 44
!
router ospf core
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
 redistribute ospf 1 route-policy 44-out
!
```

There is still a loop here for the destination 33.56.4.0/24.

Shut the G0/0/0/1 interface on R1 to see which prefix will be installed.

```
RP/0/0/CPU0:SP1R1(config)#int g0/0/0/1
RP/0/0/CPU0:SP1R1(config-if)#shut
RP/0/0/CPU0:SP1R1(config-if)#commit
```

After interface shut:

```
RP/0/0/CPU0:SP1R1#sh route 33.56.4.0
Tue Jan  1 03:05:30.331 UTC

Routing entry for 33.56.4.0/24
  Known via "ospf core", distance 110, metric 103, type intra area
  Installed Jan  1 03:04:36.085 for 00:00:54
  Routing Descriptor Blocks
    33.56.15.5, from 33.56.4.4, via GigabitEthernet0/0/0/3
      Route metric is 103
  No advertising protos.
```

Do '**no shut**' to enable this interface back again. The previous path through R6 is installed again.

```
RP/0/0/CPU0:SP1R1#sh route 33.56.4.0
Tue Jan  1 03:05:42.280 UTC

Routing entry for 33.56.4.0/24
  Known via "ospf 1", distance 110, metric 102
  Tag 44, type extern 2
  Installed Jan  1 03:05:35.350 for 00:00:06
  Routing Descriptor Blocks
    40.60.16.6, from 33.56.2.2, via GigabitEthernet0/0/0/1
      Route metric is 102
  No advertising protos.
```

As you can see the mutual redistribution is not the only issue. Once loopback0 of R4 is enabled the R1 installs the external prefix 33.56.4.0/24 from the OSPF process 1 even though there is the intra-area prefix from the OSPF process core. This is not a bug. This works as expected. How to change it? Do not install in the OSPF database prefixes already redistributed from OSPF core so with tag 44.

```
RP/0/0/CPU0:SP1R1(config)#router ospf 1
RP/0/0/CPU0:SP1R1(config-ospf)#distribute-list route-policy 44-out in
RP/0/0/CPU0:SP1R1(config-ospf)#commit

RP/0/0/CPU0:SP1R1#sh route 33.56.4.0
Tue Jan  1 03:06:59.845 UTC

Routing entry for 33.56.4.0/24
  Known via "ospf core", distance 110, metric 103, type intra area
  Installed Jan  1 03:06:50.835 for 00:00:09
  Routing Descriptor Blocks
    33.56.15.5, from 33.56.4.4, via GigabitEthernet0/0/0/3
      Route metric is 103
  No advertising protos.
```

Finally, you get rid of the loop.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R2#traceroute 33.56.4.4
Tue Jan  1 13:48:33.218 UTC

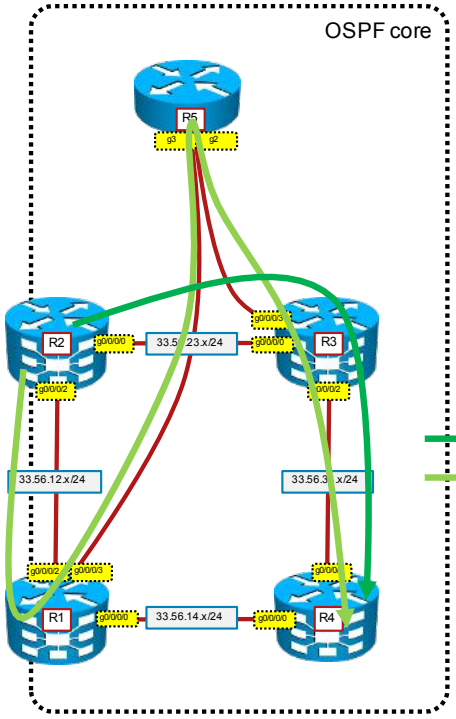
Type escape sequence to abort.
Tracing the route to 33.56.4.4

 1  33.56.12.1 0 msec  0 msec  0 msec
 2  33.56.15.5 0 msec  0 msec  0 msec
 3  33.56.35.3 0 msec  0 msec  0 msec
 4  33.56.34.4 0 msec  *  0 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4: Why ECMP is not equal?

As you could see from the previous output the ECMP on R2 to 33.56.4.0/24 is not working like ECMP even if it could.




```
RP/0/0/CPU0:SP1R2#sh route 33.56.4.4

Routing entry for 33.56.4.0/24
  Known via "ospf core", distance 110, metric 102, type inter area
  Installed Mar 21 02:53:23.551 for 11:05:03
  Routing Descriptor Blocks
    33.56.23.3, from 33.56.3.3, via GigabitEthernet0/0/0/0
      Route metric is 102
    33.56.12.1, from 33.56.4.4, via GigabitEthernet0/0/0/2
      Route metric is 102
  No advertising protos.
```

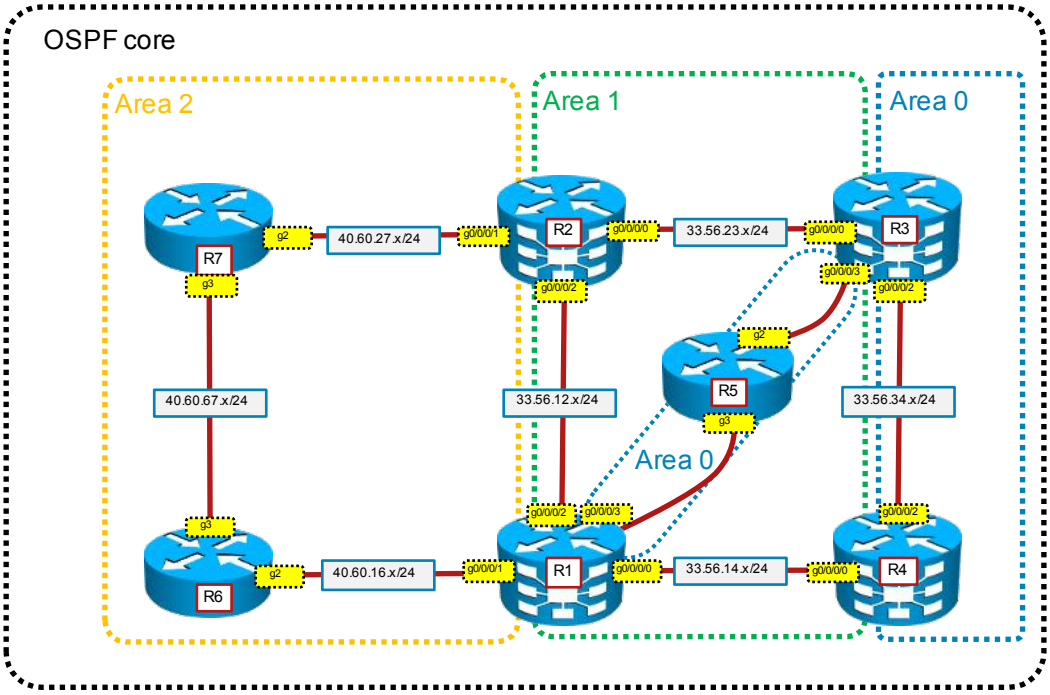
What is the reason that the second path via R1 is not going directly to R4? Is it the OSPF metric?

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

AS3356 OSPF

Prepare the OSPF configuration in AS3356.

There are two OSPF processes in AS 3356. The process named '**core**' and **1**.



Task 1: Migrate OSPF 1 to core

Migrate the OSPF process 1 to OSPF process core. Include interfaces from the process 1 to the core area 2 according to the topology depicted in the chapter.

```
RP/0/0/CPU0:SP1R2(config)#no router ospf 1
RP/0/0/CPU0:SP1R2(config)#router ospf core
RP/0/0/CPU0:SP1R2(config-ospf)#area 2
RP/0/0/CPU0:SP1R2(config-ospf-ar)#int g0/0/0/1
RP/0/0/CPU0:SP1R2(config-ospf-ar-if)#commit

RP/0/0/CPU0:SP1R1(config)#no router ospf 1
RP/0/0/CPU0:SP1R1(config)#router ospf core
RP/0/0/CPU0:SP1R1(config-ospf)#area 2
RP/0/0/CPU0:SP1R1(config-ospf-ar)#int g0/0/0/1
RP/0/0/CPU0:SP1R1(config-ospf-ar-if)#commit

CEL6(config)#router ospf 1
CEL6(config-router)#no network 0.0.0.0 255.255.255.255 area 0
CEL6(config-router)#network 0.0.0.0 255.255.255.255 area 2

CEL7(config)#router ospf 1
CEL7(config-router)#no network 0.0.0.0 255.255.255.255 area 0
CEL7(config-router)#network 0.0.0.0 255.255.255.255 area 2
```

Verification in area 2.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CEL7#sh ip protocols

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 172.16.77.77
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    0.0.0.0 255.255.255.255 area 2
  Routing Information Sources:
    Gateway         Distance      Last Update
    40.60.16.1           110      1d17h
    33.56.14.1           110      00:06:33
    33.56.4.4            110      00:06:33
    33.56.1.1            110      14:05:19
    33.56.2.2            110      01:18:20
    172.16.6.6           110      00:06:33
  Distance: (default is 110)


CEL7#sh ip ospf nei

Neighbor ID     Pri   State           Dead Time   Address         Interface
172.16.6.6        1   FULL/BDR        00:00:33    44.44.67.6      GigabitEthernet3
33.56.23.2        1   FULL/DR         00:00:34    40.60.27.2      GigabitEthernet2

CEL7#sh ip route summary
IP routing table name is default (0x0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Replicates  Overhead    Memory (bytes)
application     0           0           0           0           0
connected       0           8           0           768         2368
static          0           0           0           0           0
isis core       0           0           0           0           0
  Level 1: 0 Level 2: 0 Inter-area: 0
ospf 1          0           18          0           1728        5400
  Intra-area: 3 Inter-area: 12 External-1: 0 External-2: 3
  NSSA External-1: 0 NSSA External-2: 0
internal        9                                               5384
Total           9           26          0           2496        13152
```

ABR R1 verification.

```
RP/0/0/CPU0:SP1R1#sh protocols ospf
Tue Jan  1 17:14:17.712 UTC

Routing Protocol OSPF core
  Router Id: 33.56.14.1
  Distance: 110
  Non-Stop Forwarding: Disabled
  Redistribution:
    None
  Area 0
    Loopback0
    GigabitEthernet0/0/0/3
  Area 1
    GigabitEthernet0/0/0/0
    GigabitEthernet0/0/0/2
  Area 2
    GigabitEthernet0/0/0/1

RP/0/0/CPU0:SP1R1#sh route ospf
Tue Jan  1 17:14:59.979 UTC

O E2 10.0.128.0/29 [110/20] via 33.56.14.4, 14:21:37, GigabitEthernet0/0/0/0
O E2 11.11.44.0/24 [110/20] via 33.56.14.4, 14:21:37, GigabitEthernet0/0/0/0
O    21.41.46.0/24 [110/2] via 41.61.16.6, 00:10:58, GigabitEthernet0/0/0/1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
O    22.42.37.0/24 [110/3] via 41.61.16.6, 00:10:58, GigabitEthernet0/0/0/1
O    33.56.2.2/32 [110/2] via 33.56.12.2, 14:08:09, GigabitEthernet0/0/0/2
O    33.56.3.3/32 [110/102] via 33.56.15.5, 14:08:09, GigabitEthernet0/0/0/3
O    33.56.4.4/32 [110/103] via 33.56.15.5, 01:56:20, GigabitEthernet0/0/0/3
O    33.56.5.5/32 [110/101] via 33.56.15.5, 14:21:49, GigabitEthernet0/0/0/3
O    33.56.23.0/24 [110/101] via 33.56.12.2, 14:08:09, GigabitEthernet0/0/0/2
O    33.56.34.0/24 [110/102] via 33.56.15.5, 14:08:09, GigabitEthernet0/0/0/3
O    33.56.35.0/24 [110/101] via 33.56.15.5, 14:21:49, GigabitEthernet0/0/0/3
O    33.56.44.44/32 [110/103] via 33.56.15.5, 14:08:09, GigabitEthernet0/0/0/3
O    40.60.27.0/24 [110/3] via 41.61.16.6, 00:10:58, GigabitEthernet0/0/0/1
O E2 41.61.46.0/24 [110/20] via 33.56.14.4, 14:21:37, GigabitEthernet0/0/0/0
O    44.44.67.0/24 [110/2] via 41.61.16.6, 00:10:58, GigabitEthernet0/0/0/1
O    172.16.6.6/32 [110/2] via 41.61.16.6, 00:10:58, GigabitEthernet0/0/0/1
O    172.16.7.7/32 [110/3] via 41.61.16.6, 00:10:58, GigabitEthernet0/0/0/1
O    172.16.77.77/32 [110/3] via 41.61.16.6, 00:10:58, GigabitEthernet0/0/0/1


RP/0/0/CPU0:SP1R1#sh ospf data summ 172.16.7.7

            OSPF Router with ID (33.56.14.1) (Process ID core)

                Summary Net Link States (Area 0)

  LS age: 783
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.7 (Summary Network Number)
  Advertising Router: 33.56.14.1
  LS Seq Number: 80000001
  Checksum: 0x3acc
  Length: 28
  Network Mask: /32
        TOS: 0  Metric: 3


                Summary Net Link States (Area 1)

  LS age: 785
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.7 (Summary Network Number)
  Advertising Router: 33.56.3.3
  LS Seq Number: 80000001
  Checksum: 0x8f7e
  Length: 28
  Network Mask: /32
        TOS: 0  Metric: 5

  LS age: 784
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.7 (Summary Network Number)
  Advertising Router: 33.56.4.4
  LS Seq Number: 80000001
  Checksum: 0x8c7e
  Length: 28
  Network Mask: /32
        TOS: 0  Metric: 6

  LS age: 783
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.7 (Summary Network Number)
  Advertising Router: 33.56.14.1
  LS Seq Number: 80000001
  Checksum: 0x3acc
  Length: 28
  Network Mask: /32
        TOS: 0  Metric: 3
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Inter-area communication

For the sake of this task check R2's router-id. If it is not 33.56.23.2 please set this up manually be the 'router-id' command.

As you can see there is no R2 on the list of ABRs. Why?

```
RP/0/0/CPU0:SP1R2#sh ospf data summ 172.16.7.7

            OSPF Router with ID (33.56.23.2) (Process ID core)
                Summary Net Link States (Area 1)

  LS age: 1045
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.7 (Summary Network Number)
  Advertising Router: 33.56.3.3
  LS Seq Number: 80000001
  Checksum: 0x8f7e
  Length: 28
  Network Mask: /32
        TOS: 0  Metric: 5

  LS age: 1046
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.7 (Summary Network Number)
  Advertising Router: 33.56.4.4
  LS Seq Number: 80000001
  Checksum: 0x8c7e
  Length: 28
  Network Mask: /32
        TOS: 0  Metric: 6

  LS age: 1045
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.7 (Summary Network Number)
  Advertising Router: 33.56.14.1
  LS Seq Number: 80000001
  Checksum: 0x3acc
  Length: 28
  Network Mask: /32
        TOS: 0  Metric: 3
```

The inter-area communication between area 2 and the rest of the network is going through R1 only. This is a single point of a potential failure.

```
CEL7#trace 33.56.4.4
Type escape sequence to abort.
Tracing the route to 33.56.4.4
VRF info: (vrf in name/id, vrf out name/id)
  1 44.44.67.6 2 msec 1 msec 2 msec
  2 41.61.16.1 3 msec 2 msec 3 msec
  3 33.56.15.5 3 msec 4 msec 4 msec
  4 33.56.35.3 4 msec 5 msec 5 msec
  5 33.56.34.4 10 msec *  7 msec

CEL7#sh ip route 33.56.4.4
Routing entry for 33.56.4.4/32
  Known via "ospf 1", distance 110, metric 204, type inter area
  Last update from 44.44.67.6 on GigabitEthernet3, 00:07:29 ago
  Routing Descriptor Blocks:
  * 44.44.67.6, from 33.56.14.1, 00:07:29 ago, via GigabitEthernet3
      Route metric is 204, traffic share count is 1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Every non-backbone area must be connected to area 0. There is no area 0 on R2. Use a virtual-link spanned between R2 and ABR connected to area 0 so R1, R3 or R4. For the sake of redundancy let's use R3. Configure a virtual-link between router ID of R2 and R3.

Here is R2's and R3's router ID.

```
RP/0/0/CPU0:SP1R2#sh protocols ospf

Routing Protocol OSPF core
  Router Id: 33.56.23.2


RP/0/0/CPU0:SP1R3#sh protocols ospf

Routing Protocol OSPF core
  Router Id: 33.56.3.3
```

Configuration:

```
RP/0/0/CPU0:SP1R2(config)#router ospf core
RP/0/0/CPU0:SP1R2(config-ospf)#area 1
RP/0/0/CPU0:SP1R2(config-ospf-ar)#virtual-link 33.56.3.3

RP/0/0/CPU0:SP1R3(config)#router ospf core
RP/0/0/CPU0:SP1R3(config-ospf)#area 1
RP/0/0/CPU0:SP1R3(config-ospf-ar)#virtual-link 33.56.23.2
```

Verification:

```
RP/0/0/CPU0:SP1R2#sh ospf nei

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF core

Neighbor ID     Pri   State           Dead Time   Address         Interface
33.56.3.3       1     FULL/  -        00:00:17    33.56.23.3      OSPF_VL0
    Neighbor is up for 00:00:13
33.56.3.3       1     FULL/  -        00:00:36    33.56.23.3      GigabitEthernet0/0/0/0
    Neighbor is up for 14:44:41
33.56.14.1      1     FULL/BDR        00:00:30    33.56.12.1      GigabitEthernet0/0/0/2
    Neighbor is up for 14:43:04
172.16.77.77    1     FULL/BDR        00:00:38    40.60.27.7      GigabitEthernet0/0/0/1
    Neighbor is up for 00:39:19

Total neighbor count: 4
```

R2 is on the list of ABRs.

```
RP/0/0/CPU0:SP1R2#sh ospf data summ 172.16.7.7

            OSPF Router with ID (33.56.23.2) (Process ID core)

                Summary Net Link States (Area 0)

  LS age: 11 (DoNotAge)
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.7 (Summary Network Number)
  Advertising Router: 33.56.14.1
  LS Seq Number: 80000002
  Checksum: 0x38cd
  Length: 28
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Network Mask: /32
        TOS: 0  Metric: 3

  LS age: 170
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.7 (Summary Network Number)
  Advertising Router: 33.56.23.2
  LS Seq Number: 80000001
  Checksum: 0xea13
  Length: 28
  Network Mask: /32
        TOS: 0  Metric: 2


                Summary Net Link States (Area 1)

  LS age: 148
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.7 (Summary Network Number)
  Advertising Router: 33.56.3.3
  LS Seq Number: 80000004
  Checksum: 0x7597
  Length: 28
  Network Mask: /32
        TOS: 0  Metric: 3

  LS age: 162
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.7 (Summary Network Number)
  Advertising Router: 33.56.4.4
  LS Seq Number: 80000003
  Checksum: 0x7496
  Length: 28
  Network Mask: /32
        TOS: 0  Metric: 4

  LS age: 140
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.7 (Summary Network Number)
  Advertising Router: 33.56.14.1
  LS Seq Number: 80000003
  Checksum: 0x36ce
  Length: 28
  Network Mask: /32
        TOS: 0  Metric: 3

  LS age: 170
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.7 (Summary Network Number)
  Advertising Router: 33.56.23.2
  LS Seq Number: 80000001
  Checksum: 0xea13
  Length: 28
  Network Mask: /32
        TOS: 0  Metric: 2
```

The traffic is going via the nearest exit from the area 2.

```
CEL7#trace 33.56.4.4
Type escape sequence to abort.
Tracing the route to 33.56.4.4
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 3 msec 1 msec 2 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  2 33.56.23.3 3 msec 3 msec 4 msec
  3 33.56.34.4 7 msec *  6 msec

CEL7#sh ip route 33.56.4.4
Routing entry for 33.56.4.4/32
  Known via "ospf 1", distance 110, metric 103, type inter area
  Last update from 40.60.27.2 on GigabitEthernet2, 00:04:30 ago
  Routing Descriptor Blocks:
  * 40.60.27.2, from 33.56.23.2, 00:04:30 ago, via GigabitEthernet2
      Route metric is 103, traffic share count is 1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3: Scalability and path optimization

As the result of many ABRs in this small network there are many LSAs Type-3 (Summary LSA).

```
RP/0/0/CPU0:SP1R1#sh ospf data database-summary

            OSPF Router with ID (33.56.14.1) (Process ID core)

Area 0 database summary
  LSA Type      Count    Delete   Maxage
  Router        5        0        0
  Network       3        0        0
  Summary Net   32       0        0
  Summary ASBR  6        0        0
  Type-7 Ext    0        0        0
  Opaque Link   0        0        0
  Opaque Area   0        0        0
  Subtotal      46       0        0

Area 1 database summary
  LSA Type      Count    Delete   Maxage
  Router        4        0        0
  Network       2        0        0
  Summary Net   64       0        0
  Summary ASBR  6        0        0
  Type-7 Ext    0        0        0
  Opaque Link   0        0        0
  Opaque Area   0        0        0
  Subtotal      76       0        0

Area 2 database summary
  LSA Type      Count    Delete   Maxage
  Router        4        0        0
  Network       3        0        0
  Summary Net   24       0        0
  Summary ASBR  3        0        0
  Type-7 Ext    0        0        0
  Opaque Link   0        0        0
  Opaque Area   0        0        0
  Subtotal      34       0        0

Process core database summary
  Router        13       0        0
  Network       8        0        0
  Summary Net   120      0        0
  Summary ASBR  15       0        0
  Type-7 Ext    0        0        0
  Opaque Link   0        0        0
  Opaque Area   0        0        0
  Type-5 Ext    5        0        0
  Opaque AS     0        0        0
  Total         161      0        0
```

There are 13 Router LSAs. Do you know why 13 and not just 7 which is the number of routers in OSPF core? You need to count routers for each of area separately. You can't minimize the number of Router LSAs, but you can change the number of Network LSAs, advertised prefixes and as a consequence Summary LSAs.

Enable prefix-suppression for each OSPF process, change the network type of interfaces to point-to-point to avoid DR/BDR election, do not advertise external LSAs as the prefixes are learnt internally.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Advertise just Loopbacks or host routes which will be used by L2/L3MPLS VPN services.

Example of configuration on R2:

```
RP/0/0/CPU0:SP1R2#sh run router ospf
Tue Jan  1 18:35:27.839 UTC
router ospf core
 prefix-suppression
 address-family ipv4 unicast
 area 1
  virtual-link 33.56.3.3
  !
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
   cost 100
   network point-to-point
  !
  interface GigabitEthernet0/0/0/2
   network point-to-point
  !
 !
 area 2
  interface GigabitEthernet0/0/0/1
   network point-to-point
  !
 !
!
```

The 'network point-to-point' minimizes number of Network LSAs by getting rid of DRs and BDRs.

```
RP/0/0/CPU0:SP1R1(config)#router ospf core
RP/0/0/CPU0:SP1R1(config-ospf)#prefix-suppression
RP/0/0/CPU0:SP1R1(config-ospf)#area 0
RP/0/0/CPU0:SP1R1(config-ospf-ar)#int g0/0/0/3
RP/0/0/CPU0:SP1R1(config-ospf-ar-if)#network point-to-p
RP/0/0/CPU0:SP1R1(config-ospf-ar-if)#area 1
RP/0/0/CPU0:SP1R1(config-ospf-ar)#int g0/0/0/0
RP/0/0/CPU0:SP1R1(config-ospf-ar-if)#network point-to-p
RP/0/0/CPU0:SP1R1(config-ospf-ar-if)#int g0/0/0/2
RP/0/0/CPU0:SP1R1(config-ospf-ar-if)#network point-to-p
RP/0/0/CPU0:SP1R1(config-ospf-ar-if)#area 2
RP/0/0/CPU0:SP1R1(config-ospf-ar)#int g0/0/0/1
RP/0/0/CPU0:SP1R1(config-ospf-ar-if)#network point-to-p
```

Example of configuration on R7.

```
CEL7(config)#router ospf 1
CEL7(config-router)#prefix-suppression
CEL7(config-router)#int g2
CEL7(config-if)#ip ospf net point-to-p
CEL7(config-if)#int g3
CEL7(config-if)#ip ospf net point-to-p
```

Make sure that you see all 9 Loopbacks

```
SP1-R5-RR(config)#int lo0
SP1-R5-RR(config-if)#ip ospf prefix-suppression disable
```

After optimization.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R1#sh route

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

Gateway of last resort is not set

L     33.56.1.1/32 is directly connected, 18:58:46, Loopback0
O     33.56.2.2/32 [110/2] via 33.56.12.2, 00:46:26, GigabitEthernet0/0/0/2
O     33.56.3.3/32 [110/102] via 33.56.15.5, 00:13:50, GigabitEthernet0/0/0/3
O     33.56.4.4/32 [110/103] via 33.56.15.5, 00:13:46, GigabitEthernet0/0/0/3
O     33.56.5.5/32 [110/101] via 33.56.15.5, 00:03:50, GigabitEthernet0/0/0/3
C     33.56.12.0/24 is directly connected, 1d19h, GigabitEthernet0/0/0/2
L     33.56.12.1/32 is directly connected, 1d19h, GigabitEthernet0/0/0/2
C     33.56.14.0/24 is directly connected, 1d16h, GigabitEthernet0/0/0/0
L     33.56.14.1/32 is directly connected, 1d16h, GigabitEthernet0/0/0/0
C     33.56.15.0/24 is directly connected, 1d19h, GigabitEthernet0/0/0/3
L     33.56.15.1/32 is directly connected, 1d19h, GigabitEthernet0/0/0/3
O     33.56.44.44/32 [110/103] via 33.56.15.5, 00:13:46, GigabitEthernet0/0/0/3
C     41.61.16.0/24 is directly connected, 02:25:09, GigabitEthernet0/0/0/1
L     41.61.16.1/32 is directly connected, 02:25:09, GigabitEthernet0/0/0/1
O     172.16.6.6/32 [110/2] via 41.61.16.6, 00:46:30, GigabitEthernet0/0/0/1
O     172.16.7.7/32 [110/3] via 41.61.16.6, 00:46:30, GigabitEthernet0/0/0/1
O     172.16.77.77/32 [110/3] via 41.61.16.6, 00:01:27, GigabitEthernet0/0/0/1


RP/0/0/CPU0:SP1R1#sh ospf data database-summary

            OSPF Router with ID (33.56.14.1) (Process ID core)

Area 0 database summary
  LSA Type      Count    Delete   Maxage
  Router        5        0        0
  Network       0        0        0
  Summary Net   10       0        0
  Summary ASBR  0        0        0
  Type-7 Ext    0        0        0
  Opaque Link   0        0        0
  Opaque Area   0        0        0
  Subtotal      15       0        0

Area 1 database summary
  LSA Type      Count    Delete   Maxage
  Router        4        0        0
  Network       0        0        0
  Summary Net   25       0        0
  Summary ASBR  0        0        0
  Type-7 Ext    0        0        0
  Opaque Link   0        0        0
  Opaque Area   0        0        0
  Subtotal      29       0        0

Area 2 database summary
  LSA Type      Count    Delete   Maxage
  Router        4        0        0
  Network       0        0        0
  Summary Net   8        0        0
  Summary ASBR  0        0        0
  Type-7 Ext    0        0        0
  Opaque Link   0        0        0
  Opaque Area   0        0        0
  Subtotal      12       0        0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Process core database summary
  Router         13        0         0
  Network        0         0         0
  Summary Net    43        0         0
  Summary ASBR   0         0         0
  Type-7 Ext     0         0         0
  Opaque Link    0         0         0
  Opaque Area    0         0         0
  Type-5 Ext     0         0         0
  Opaque AS      0         0         0
  Total          56        0         0
```

Memory consumption before configuration:

```
RP/0/0/CPU0:SP1R1#sh route summary

Route Source                     Routes     Backup     Deleted     Memory(bytes)
local                            5          0          0           800
connected                        4          1          0           800
ospf core                        18         0          0           2880
dagr                             0          0          0           0
Total                            27         1          0           4480
```

Memory consumption after configuration:

```
RP/0/0/CPU0:SP1R1#sh route summ

Route Source                     Routes     Backup     Deleted     Memory(bytes)
local                            5          0          0           800
connected                        4          1          0           800
ospf core                        8          0          0           1280
dagr                             0          0          0           0
Total                            17         1          0           2880
```

Now the virtual link is down:

```
RP/0/0/CPU0:SP1R3#RP/0/0/CPU0:Jan 11 20:05:10.336 : ospf[1018]: %ROUTING-OSPF-4-ERRRCV :
Received invalid packet: mismatch area ID, from backbone area must be virtual-link but
not found from 33.56.23.2, GigabitEthernet0/0/0/0
```

Router-id of R2 is not reachable in the OSPF database.

```
RP/0/0/CPU0:SP1R3#show ip ospf core 0 database router 33.56.23.2

            OSPF Router with ID (33.56.3.3) (Process ID core)

                Router Link States (Area 0)

  Adv Router is not-reachable
  LS age: 1 (DoNotAge)
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 33.56.23.2
  Advertising Router: 33.56.23.2
  LS Seq Number: 8000000f
  Checksum: 0x2bdc
  Length: 36
  Area Border Router
   Number of Links: 1

    Link connected to: a Virtual Link
     (Link ID) Neighboring Router ID: 33.56.3.3
     (Link Data) Router Interface address: 33.56.23.2
      Number of TOS metrics: 0
       TOS 0 Metrics: 100
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

You need to populate R2's the reachability of 'Router Interface address: 33.56.23.2' by disabling prefix-suppression on G0/0/0/0 of R2 or by changing router-id.

```
RP/0/0/CPU0:SP1R2(config)#router ospf core
RP/0/0/CPU0:SP1R2(config-ospf)#area 1
RP/0/0/CPU0:SP1R2(config-ospf-ar-if)#interface GigabitEthernet0/0/0/0
RP/0/0/CPU0:SP1R2(config-ospf-ar-if)#prefix-suppression disable

RP/0/0/CPU0:SP1R3#sh ospf nei

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF core

Neighbor ID     Pri   State           Dead Time   Address         Interface
33.56.23.2      1     FULL/  -        00:00:13    33.56.23.2      OSPF_VL0
    Neighbor is up for 00:00:26
[...]
Total neighbor count: 4

RP/0/0/CPU0:SP1R3#show ip ospf core 0 database router 33.56.23.2


            OSPF Router with ID (33.56.3.3) (Process ID core)

                Router Link States (Area 0)

  Routing Bit Set on this LSA
  LS age: 1 (DoNotAge)
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 33.56.23.2
  Advertising Router: 33.56.23.2
  LS Seq Number: 80000014
  Checksum: 0x21e1
  Length: 36
  Area Border Router
   Number of Links: 1

    Link connected to: a Virtual Link
     (Link ID) Neighboring Router ID: 33.56.3.3
     (Link Data) Router Interface address: 33.56.23.2
      Number of TOS metrics: 0
       TOS 0 Metrics: 100
```

To see a clear distinction between router-id and the router interface you can change the IP address of G0/0/0/0 on R2.

```
RP/0/0/CPU0:SP1R2(config)#int g0/0/0/0
RP/0/0/CPU0:SP1R2(config-if)#ipv4 add 33.56.23.22/24

RP/0/0/CPU0:SP1R3#clear ospf process

Reset ALL OSPF processes? [no]: yes

RP/0/0/CPU0:SP1R3#show ip ospf core 0 database router 33.56.23.2

            OSPF Router with ID (33.56.3.3) (Process ID core)

                Router Link States (Area 0)

  Routing Bit Set on this LSA
  LS age: 1 (DoNotAge)
  Options: (No TOS-capability, DC)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  LS Type: Router Links
  Link State ID: 33.56.23.2
  Advertising Router: 33.56.23.2
  LS Seq Number: 8000001a
  Checksum: 0x2eba
  Length: 36
  Area Border Router
   Number of Links: 1

    Link connected to: a Virtual Link
     (Link ID) Neighboring Router ID: 33.56.3.3
     (Link Data) Router Interface address: 33.56.23.22
      Number of TOS metrics: 0
       TOS 0 Metrics: 100
```

Revert back the previous IP address by the 'rollback configuration last 1' command.

The last part of this task is path optimization. Let's assume that the link between R6 and R7 is lower quality than the rest of the network. You need to redirect the traffic via R2 and R1.

The current state:

```
CEL7#traceroute 172.16.6.6 source 172.16.7.7
Type escape sequence to abort.
Tracing the route to 172.16.6.6
VRF info: (vrf in name/id, vrf out name/id)
  1 44.44.67.6 2 msec *  2 msec
```

The cost change is not enough as intra-area routes are preferred over inter-area.

```
CEL7#sh ip route | i 172.16.6.6
O        172.16.6.6 [110/2] via 44.44.67.6, 00:33:33, GigabitEthernet3


CEL7(config)#int g3
CEL7(config-if)#ip ospf cost 300


CEL7# sh route | i 172.16.6.6
O        172.16.6.6 [110/301] via 44.44.67.6, 00:00:02, GigabitEthernet3
```

You can't migrate the link between R1 and R2 to area 2 as the other traffic going through area 1wouldn't be optimal. You can configure the R1-R2 link as multi-area.

```
RP/0/0/CPU0:SP1R2#sh ospf core 2 data router 33.56.23.2

            OSPF Router with ID (33.56.23.2) (Process ID core)

                Router Link States (Area 2)

  LS age: 964
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 33.56.23.2
  Advertising Router: 33.56.23.2
  LS Seq Number: 8000001b
  Checksum: 0xadad
  Length: 36
  Area Border Router
   Number of Links: 1

    Link connected to: another Router (point-to-point)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
     (Link ID) Neighboring Router ID: 172.16.77.77
     (Link Data) Router Interface address: 40.60.27.2
      Number of TOS metrics: 0
       TOS 0 Metrics: 1


RP/0/0/CPU0:SP1R1(config)#router ospf core
RP/0/0/CPU0:SP1R1(config-ospf)#area 2
RP/0/0/CPU0:SP1R1(config-ospf-ar)#multi-area-interface GigabitEthernet0/0/0/2

RP/0/0/CPU0:SP1R2(config)#router ospf core
RP/0/0/CPU0:SP1R2(config-ospf)#area 2
RP/0/0/CPU0:SP1R2(config-ospf-ar)#multi-area-interface GigabitEthernet0/0/0/2

RP/0/0/CPU0:SP1R1#sh ospf int brief

* Indicates MADJ interface, (P) Indicates fast detect hold down state

Interfaces for OSPF core

Interface          PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Lo0                core  0               33.56.1.1/32       1     LOOP  0/0
Gi0/0/0/3          core  0               33.56.15.1/24      100   P2P   1/1
Gi0/0/0/0          core  1               33.56.14.1/24      100   P2P   1/1
Gi0/0/0/2          core  1               33.56.12.1/24      1     P2P   1/1
Gi0/0/0/2*         core  2               33.56.12.1/24      1     P2P   0/0
Gi0/0/0/1          core  2               41.61.16.1/24      1     P2P   1/1

RP/0/0/CPU0:SP1R1#sh ospf nei

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF core

Neighbor ID     Pri   State           Dead Time   Address         Interface
33.56.5.5       1     FULL/  -        00:00:30    33.56.15.5      GigabitEthernet0/0/0/3
    Neighbor is up for 03:22:13
33.56.4.4       1     FULL/  -        00:00:38    33.56.14.4      GigabitEthernet0/0/0/0
    Neighbor is up for 03:21:07
33.56.23.2      1     FULL/  -        00:00:37    33.56.12.2      GigabitEthernet0/0/0/2
    Neighbor is up for 00:07:20
33.56.23.2      1     FULL/  -        00:00:30    33.56.12.2      GigabitEthernet0/0/0/2*
    Neighbor is up for 00:02:12
172.16.6.6      1     FULL/  -        00:00:34    41.61.16.6      GigabitEthernet0/0/0/1
    Neighbor is up for 03:32:16

Total neighbor count: 5

RP/0/0/CPU0:SP1R2#sh ospf core 2 data router 33.56.23.2

            OSPF Router with ID (33.56.23.2) (Process ID core)

                Router Link States (Area 2)

  LS age: 153
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 33.56.23.2
  Advertising Router: 33.56.23.2
  LS Seq Number: 80000007
  Checksum: 0x6896
  Length: 48
  Area Border Router
   Number of Links: 2

    Link connected to: another Router (point-to-point)
     (Link ID) Neighboring Router ID: 33.56.1.1 <- clear ospf process changed ID.
     (Link Data) Router Interface address: 0.0.0.6
      Number of TOS metrics: 0
       TOS 0 Metrics: 1

    Link connected to: another Router (point-to-point)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
     (Link ID) Neighboring Router ID: 172.16.77.77
     (Link Data) Router Interface address: 40.60.27.2
      Number of TOS metrics: 0
       TOS 0 Metrics: 1


CEL7#sh ip route | i 172.16.6.6
O        172.16.6.6 [110/4] via 40.60.27.2, 00:03:53, GigabitEthernet2

CEL7#traceroute 172.16.6.6 source 172.16.7.7
Type escape sequence to abort.
Tracing the route to 172.16.6.6
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 2 msec 1 msec 1 msec
  2 33.56.12.1 2 msec 2 msec 2 msec
  3 41.61.16.6 3 msec *  18 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4: Secure the OSPF domain

Turn on the OSPF authentication in the area 0 and 1.

```
RP/0/0/CPU0:SP1R2#sh run key chain

key chain ospf-keys
 key 1
  accept-lifetime 00:00:00 january 01 2017 infinite
  key-string password 094F471A1A0A
  send-lifetime 00:00:00 january 01 2017 infinite
  cryptographic-algorithm MD5
 !
 accept-tolerance infinite
!

SP1-R5-RR#sh run | section key chain
key chain ospf-keys
 key 1
  key-string cisco
   cryptographic-algorithm md5
```

XR:

```
!
router ospf core
 area 0
  authentication message-digest keychain ospf-keys
!
 area 1
  authentication message-digest keychain ospf-keys
!
```

IOS XE (R5):

```
!
interface GigabitEthernet2
 ip ospf authentication key-chain ospf-keys
 ip ospf network point-to-point
!
interface GigabitEthernet3
 ip ospf authentication key-chain ospf-keys
 ip ospf network point-to-point
!
!
router ospf 1
 area 0 authentication message-digest
!
```

Verify the key validity.

```
RP/0/0/CPU0:SP1R1#show key chain ospf-keys

Key-chain: ospf-keys -

accept-tolerance -- infinite
  Key 1 -- text "094F471A1A0A"
    Cryptographic-Algorithm -- MD5
    Send lifetime --  00:00:00, 01 Jan 2017 - Always valid  [Valid now]
    Accept lifetime -- 00:00:00, 01 Jan 2017 - Always valid [Valid now]
```

Verify OSPF neighbors and routes.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R1#sh protocols ospf

Routing Protocol OSPF core
  Router Id: 33.56.1.1
  Distance: 110
  Non-Stop Forwarding: Disabled
  Redistribution:
    None
  Area 0
    Loopback0
      authentication md5
    GigabitEthernet0/0/0/3
      authentication md5
  Area 1
    GigabitEthernet0/0/0/0
      authentication md5
    GigabitEthernet0/0/0/2
      authentication md5
  Area 2
    GigabitEthernet0/0/0/2
    GigabitEthernet0/0/0/1


RP/0/0/CPU0:SP1R3#sh ospf nei

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF core

Neighbor ID     Pri   State           Dead Time   Address         Interface
33.56.23.2      1     FULL/  -        00:00:29    33.56.23.2      OSPF_VL0
    Neighbor is up for 00:00:10
33.56.4.4       1     FULL/  -        00:00:38    33.56.34.4      GigabitEthernet0/0/0/2
    Neighbor is up for 00:08:18
33.56.5.5       1     FULL/  -        00:00:37    33.56.35.5      GigabitEthernet0/0/0/3
    Neighbor is up for 00:08:22
33.56.23.2      1     FULL/  -        00:00:35    33.56.23.2      GigabitEthernet0/0/0/0
    Neighbor is up for 00:00:20

Total neighbor count: 4

RP/0/0/CPU0:SP1R1#sh ospf nei

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF core

Neighbor ID     Pri   State           Dead Time   Address         Interface
33.56.5.5       1     FULL/  -        00:00:37    33.56.15.5      GigabitEthernet0/0/0/3
    Neighbor is up for 00:06:49
33.56.4.4       1     FULL/  -        00:00:33    33.56.14.4      GigabitEthernet0/0/0/0
    Neighbor is up for 00:07:53
33.56.23.2      1     FULL/  -        00:00:31    33.56.12.2      GigabitEthernet0/0/0/2
    Neighbor is up for 00:00:43
33.56.23.2      1     FULL/  -        00:00:32    33.56.12.2
GigabitEthernet0/0/0/2*
    Neighbor is up for 01:00:08
172.16.6.6      1     FULL/  -        00:00:39    41.61.16.6      GigabitEthernet0/0/0/1
    Neighbor is up for 01:00:12

Total neighbor count: 5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

AS3356 OSPFv3

Deploy OSPFv3 configuration in AS3356.

Task 1: Configure OSPFv3 areas

Configure area 0, 1 and 2 along with the IPv4 topology.

Useful configuration commands for XR are generally the same as for OSPFv2. You can copy and paste the configuration. Some features are not supported like multi-area link.

```
RP/0/0/CPU0:SP1R1(config-ospfv3-ar-if)#sh config
Building configuration...
!! IOS XR Configuration 6.1.2
router ospfv3 core
 prefix-suppression
 area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/3
   cost 100
   network point-to-point
  !
 !
 area 1
  interface GigabitEthernet0/0/0/0
   cost 100
   network point-to-point
  !
  interface GigabitEthernet0/0/0/2
   network point-to-point
  !
 !
 area 2
  interface GigabitEthernet0/0/0/1
   network point-to-point
  !
 !
!
```

IOS XE (R5):

```
SP1-R5-RR(config)#router ospfv3 1
SP1-R5-RR(config-router)#prefix-suppression
SP1-R5-RR(config-if)#int g2
SP1-R5-RR(config-if)#ipv6 ospf 1 area 0
SP1-R5-RR(config-if)#ipv6 ospf network point-to-point
SP1-R5-RR(config-if)#int g3
SP1-R5-RR(config-if)#ipv6 ospf 1 area 0
SP1-R5-RR(config-if)#ipv6 ospf network point-to-point
```

Verification - check all sessions and expected Loopbacks' prefixes:

```
RP/0/0/CPU0:SP1R3#sh ospfv3 nei

# Indicates Neighbor awaiting BFD session up

Neighbors for OSPFv3 core

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
33.56.23.2      1     FULL/  -        00:00:36    4               GigabitEthernet0/0/0/0
    Neighbor is up for 00:08:56
33.56.4.4       1     FULL/  -        00:00:32    6               GigabitEthernet0/0/0/2
    Neighbor is up for 00:07:26
33.56.5.5       1     FULL/  -        00:00:38    8               GigabitEthernet0/0/0/3
    Neighbor is up for 00:01:06
33.56.23.2      1     FULL/  -           -        1000000         OSPF_VL0
    Neighbor is up for 00:08:46

Total neighbor count: 4

RP/0/0/CPU0:SP1R1#sh ospfv3 neighbor

# Indicates Neighbor awaiting BFD session up

Neighbors for OSPFv3 core

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
33.56.4.4       1     FULL/  -        00:00:38    4               GigabitEthernet0/0/0/0
    Neighbor is up for 00:12:04
33.56.23.2      1     FULL/  -        00:00:33    6               GigabitEthernet0/0/0/2
    Neighbor is up for 00:15:11
172.16.6.6      1     FULL/  -        00:00:33    8               GigabitEthernet0/0/0/1
    Neighbor is up for 00:01:16
33.56.5.5       1     FULL/  -        00:00:36    9               GigabitEthernet0/0/0/3
    Neighbor is up for 00:05:42

Total neighbor count: 4

RP/0/0/CPU0:SP1R1#sh route ipv6 ospf

O    2001:22:44:ee::7/128
      [110/2] via fe80::6, 00:04:44, GigabitEthernet0/0/0/1
O    2001:33:56:23::/64
      [110/101] via fe80::2, 00:18:44, GigabitEthernet0/0/0/2
O    2001:33:56:aa::2/128
      [110/1] via fe80::2, 00:18:44, GigabitEthernet0/0/0/2
O    2001:33:56:aa::3/128
      [110/101] via fe80::5, 00:09:11, GigabitEthernet0/0/0/3
O    2001:33:56:aa::4/128
      [110/102] via fe80::5, 00:09:11, GigabitEthernet0/0/0/3
O    2001:33:56:aa::5/128
      [110/100] via fe80::5, 00:09:11, GigabitEthernet0/0/0/3
O    2001:44:66:cc::6/128
      [110/1] via fe80::6, 00:04:22, GigabitEthernet0/0/0/1
O    2001:44:66:cc::7/128
      [110/2] via fe80::6, 00:04:44, GigabitEthernet0/0/0/1

RP/0/0/CPU0:SP1R1#sh ospfv3 summary

Routing process "ospfv3 core"
Number of OSPFv3 interfaces 5
Number of OSPFv3 interfaces up 5
Number of OSPFv3 virtual interfaces up 0
Number of OSPFv3 sham-link interfaces up 0
Number of neighbors 4
Number of neighbors adjacent 4
Number of areas 3
 LSA Type          Count
 Router            : 13
 Network           : 0
 Link              : 8
 Prefix            : 8
 Inter-Area Prefix : 54
 Inter-Area Router : 0
 Type-7 Ext        : 0
 Grace             : 0
 Unknown Link      : 0
 Unknown Area      : 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
 Type-5 Ext        : 0
 Unknown AS        : 0
 Total             : 83

RP/0/0/CPU0:SP1R1#sh ospfv3 database database-summary

            OSPFv3 Router with ID (33.56.1.1) (Process ID core)

Area 0 database summary
  LSA Type            Count    Delete   Maxage   Checksum
  Router              5        0        0        0x10f36
  Network             0        0        0        0x0
  Link                2        0        0        0x16de8
  Prefix              4        0        0        0x2117c
  Inter-area Prefix   14       0        0        0x8ce84
  Inter-area Router   0        0        0        0x0
  Type-7 Ext          0        0        0        0x0
  Grace               0        0        0        0x0
  Unknown Link        0        0        0        0x0
  Unknown Area        0        0        0        0x0
  Subtotal            25       0        0        0xd5d1e

Area 1 database summary
  LSA Type            Count    Delete   Maxage   Checksum
  Router              4        0        0        0x23bab
  Network             0        0        0        0x0
  Link                4        0        0        0x21f90
  Prefix              2        0        0        0x184a0
  Inter-area Prefix   28       0        0        0x1066f6
  Inter-area Router   0        0        0        0x0
  Type-7 Ext          0        0        0        0x0
  Grace               0        0        0        0x0
  Unknown Link        0        0        0        0x0
  Unknown Area        0        0        0        0x0
  Subtotal            38       0        0        0x1646d1

Area 2 database summary
  LSA Type            Count    Delete   Maxage   Checksum
  Router              4        0        0        0x21c5f
  Network             0        0        0        0x0
  Link                2        0        0        0x18b66
  Prefix              2        0        0        0x4d15
  Inter-area Prefix   12       0        0        0x8091e
  Inter-area Router   0        0        0        0x0
  Type-7 Ext          0        0        0        0x0
  Grace               0        0        0        0x0
  Unknown Link        0        0        0        0x0
  Unknown Area        0        0        0        0x0
  Subtotal            20       0        0        0xbfdf8

Process core database summary
  LSA Type            Count    Delete   Maxage   Checksum
  Router              13       0        0        0x56740
  Network             0        0        0        0x0
  Link                8        0        0        0x518de
  Prefix              8        0        0        0x3e331
  Inter-area Prefix   54       0        0        0x213e98
  Inter-area Router   0        0        0        0x0
  Type-7 Ext          0        0        0        0x0
  Grace               0        0        0        0x0
  Unknown Link        0        0        0        0x0
  Unknown Area        0        0        0        0x0
  Type-5 Ext          0        0        0        0x0
  Unknown AS          0        0        0        0x0
  Total               83       0        0        0x2fa1e7

RP/0/0/CPU0:SP1R1#sh ospfv3 interface

GigabitEthernet0/0/0/0 is up, line protocol is up
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Link Local address fe80::1, Interface ID 4
  Area 1, Process ID core, Instance ID 0, Router ID 33.56.1.1
  Network Type POINT_TO_POINT, Cost: 100
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Addresses not advertised
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:08
  Index 0/1/1, flood queue length 0
  Next 0(0)/0(0)/0(0)
  Last flood scan length is 1, maximum is 4
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 33.56.4.4
  Suppress hello for 0 neighbor(s)
  Reference count is 3
GigabitEthernet0/0/0/1 is up, line protocol is up
  Link Local address fe80::1, Interface ID 5
  Area 2, Process ID core, Instance ID 0, Router ID 33.56.1.1
  Network Type POINT_TO_POINT, Cost: 1
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Addresses not advertised
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:06
  Index 0/1/1, flood queue length 0
  Next 0(0)/0(0)/0(0)
  Last flood scan length is 2, maximum is 2
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 172.16.6.6
  Suppress hello for 0 neighbor(s)
  Reference count is 8
GigabitEthernet0/0/0/2 is up, line protocol is up
  Link Local address fe80::1, Interface ID 6
  Area 1, Process ID core, Instance ID 0, Router ID 33.56.1.1
  Network Type POINT_TO_POINT, Cost: 1
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Addresses not advertised
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
  Index 0/2/1, flood queue length 0
  Next 0(0)/0(0)/0(0)
  Last flood scan length is 1, maximum is 5
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 33.56.23.2
  Suppress hello for 0 neighbor(s)
  Reference count is 8
GigabitEthernet0/0/0/3 is up, line protocol is up
  Link Local address fe80::1, Interface ID 7
  Area 0, Process ID core, Instance ID 0, Router ID 33.56.1.1
  Network Type POINT_TO_POINT, Cost: 100
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Addresses not advertised
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:05
  Index 0/2/1, flood queue length 0
  Next 0(0)/0(0)/0(0)
  Last flood scan length is 1, maximum is 2
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 33.56.5.5
  Suppress hello for 0 neighbor(s)
  Reference count is 12
Loopback0 is up, line protocol is up
  Link Local address fe80::c46c:eff:fed2:cb9, Interface ID 12
  Area 0, Process ID core, Instance ID 0, Router ID 33.56.1.1
  Network Type LOOPBACK, Cost: 0
  Loopback interface is treated as a stub Host
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R1#sh ospfv3 interface brief

Interface    PID        Area            IPv6 Address               Cost  State Nbrs F/C
Gi0/0/0/0    core       1               fe80::1                    100   P2P   1/1
Gi0/0/0/1    core       2               fe80::1                    1     P2P   1/1
Gi0/0/0/2    core       1               fe80::1                    1     P2P   1/1
Gi0/0/0/3    core       0               fe80::1                    100   P2P   1/1
Lo0          core       0               fe80::c46c:eff:fed2:cb9    0     LOOP  0/0

SP1-R5-RR#sh ipv6 ospf nei

            OSPFv3 Router with ID (33.56.5.5) (Process ID 1)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
33.56.1.1         0   FULL/  -        00:00:35    7               GigabitEthernet3
33.56.3.3         0   FULL/  -        00:00:38    6               GigabitEthernet2


SP1-R5-RR#sh ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "application"
IPv6 Routing Protocol is "bgp 3356"
  IGP synchronization is disabled
  Redistribution:
    None
  Distance:
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "ospf 1"
  Router ID 33.56.5.5
  Number of areas: 1 normal, 0 stub, 0 nssa
  Interfaces (Area 0):
    Loopback0
    GigabitEthernet3
    GigabitEthernet2
  Redistribution:
    None

CEL7#sh ipv6 route ospf
IPv6 Routing Table - default - 24 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
       NDr - Redirect, RL - RPL, O - OSPF Intra, OI - OSPF Inter
       OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1
       ON2 - OSPF NSSA ext 2, la - LISP alt, lr - LISP site-registrations
       ld - LISP dyn-eid, lA - LISP away, a - Application
OI  2001:33:56:23::/64 [110/101]
     via FE80::2, GigabitEthernet2
OI  2001:33:56:AA::1/128 [110/2]
     via FE80::6, GigabitEthernet3
OI  2001:33:56:AA::2/128 [110/1]
     via FE80::2, GigabitEthernet2
OI  2001:33:56:AA::3/128 [110/101]
     via FE80::2, GigabitEthernet2
OI  2001:33:56:AA::4/128 [110/102]
     via FE80::2, GigabitEthernet2
OI  2001:33:56:AA::5/128 [110/102]
     via FE80::2, GigabitEthernet2
     via FE80::6, GigabitEthernet3
O   2001:44:66:CC::6/128 [110/1]
     via FE80::6, GigabitEthernet3

CEL7#ping 2001:33:56:AA::4 source 2001:22:44:EE::7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:33:56:AA::4, timeout is 2 seconds:
Packet sent with a source address of 2001:22:44:EE::7
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/14/17 ms
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Secure the OSPFv3 domain

Turn on the OSPFv3 authentication in the area 0 and 1.

IOS XR (R3):

```
RP/0/0/CPU0:SP1R3(config)#router ospfv3 core
RP/0/0/CPU0:SP1R3(config-ospfv3)#area 0
RP/0/0/CPU0:SP1R3(config-ospfv3-ar)#authentication ipsec spi 3356 md5
1234567890abcdef1234567890abcdef
RP/0/0/CPU0:SP1R3(config-ospfv3-ar)#area 1
RP/0/0/CPU0:SP1R3(config-ospfv3-ar)#authentication ipsec spi 3356 md5
1234567890abcdef1234567890abcdef
```

IOS XE (R5):

```
SP1-R5-RR(config)#router ospfv3 1
SP1-R5-RR(config-router)#area 0 authentication ipsec spi 3356 md5
1234567890abcdef1234567890abcdef
```

Verification.

```
RP/0/0/CPU0:SP1R3#sh ospfv3 interface | utility egrep 'Gig\|Auth'
AH Authentication MD5, SPI 3356
GigabitEthernet0/0/0/0 is up, line protocol is up, ipsec is up
  AH Authentication MD5, SPI 3356
GigabitEthernet0/0/0/2 is up, line protocol is up, ipsec is up
  AH Authentication MD5, SPI 3356
GigabitEthernet0/0/0/3 is up, line protocol is up, ipsec is up
  AH Authentication MD5, SPI 3356
  AH Authentication MD5, SPI 3356

RP/0/0/CPU0:SP1R3#sh ospfv3 nei

# Indicates Neighbor awaiting BFD session up

Neighbors for OSPFv3 core

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
33.56.23.2      1     FULL/  -        00:00:38    4               GigabitEthernet0/0/0/0
    Neighbor is up for 00:02:03
33.56.4.4       1     FULL/  -        00:00:39    6               GigabitEthernet0/0/0/2
    Neighbor is up for 00:02:29
33.56.5.5       1     FULL/  -        00:00:39    8               GigabitEthernet0/0/0/3
    Neighbor is up for 00:00:05
33.56.23.2      1     FULL/  -           -        1000000         OSPF_VL0
    Neighbor is up for 00:01:52

Total neighbor count: 4
```

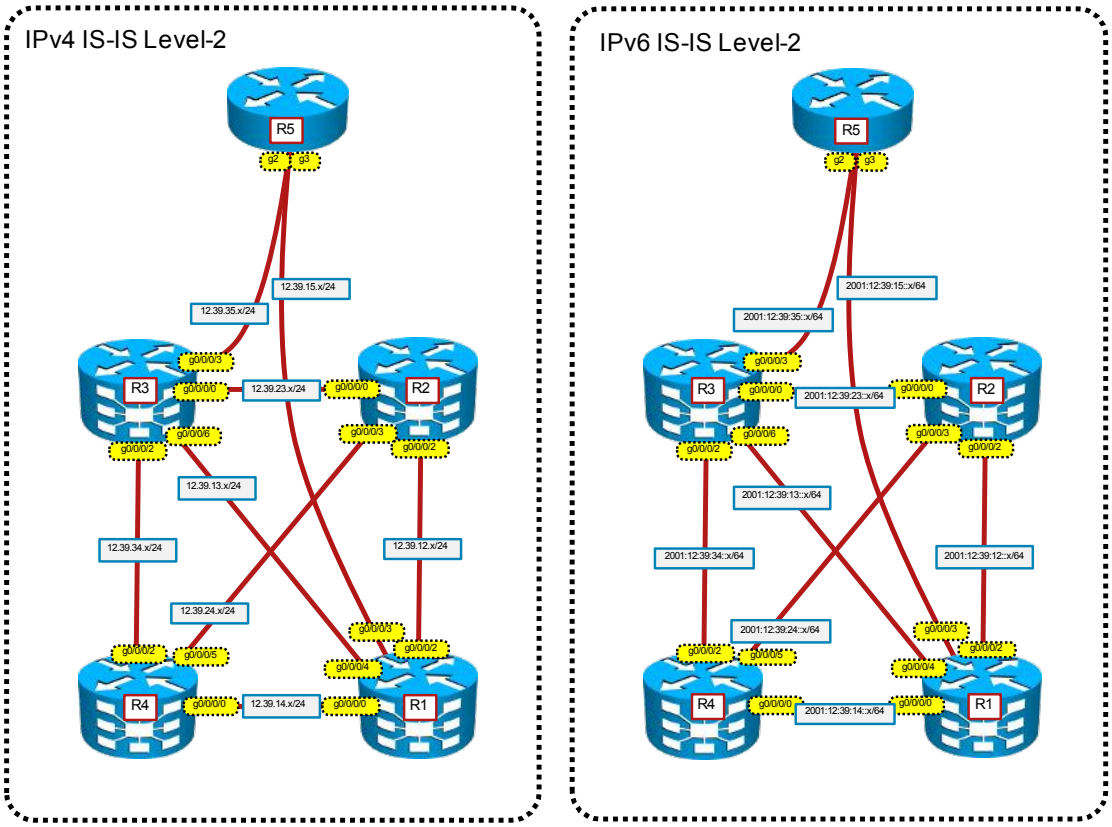

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

AS1239 ISIS

Deploy ISIS configuration in AS1239.

Topology 1: AS1239 ISIS

Configure ISIS Level-2 on routers R1-R5 in AS1239.



IPv4

Infrastructure links: 12.39.RxRy.x/24
Loopbacks: 12.39.Rx.Rx/32

IPv6

Infrastructure pool: 2001:12:39::/48

Infrastructure links: 2001:12:39:RxRy::/64

Loopbacks: 2001:12:39:FF::x/128

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: Configure ISIS Level-2

Configure Level-2 only for both IPv4 and IPv6. Run multi-topology ISIS. This is a default mode in XR but not in IOS XE.

Configure your network based on the following example.

IOS XR (R3):

```
router isis core
 is-type level-2-only
 net 1239.0000.0000.0003.00
 nsr
 address-family ipv4 unicast
  metric-style wide
  advertise passive-only
 !
 address-family ipv6 unicast
  advertise passive-only
 !
 interface Loopback0
  passive
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
  !
 !
 interface GigabitEthernet0/0/0/0
  mesh-group 1239
  point-to-point
  hello-padding sometimes
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
  !
 !
 interface GigabitEthernet0/0/0/2
  mesh-group 1239
  point-to-point
  hello-padding sometimes
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
  !
 !
 interface GigabitEthernet0/0/0/3
  point-to-point
  hello-padding sometimes
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
  !
 !
 interface GigabitEthernet0/0/0/6
  mesh-group 1239
  point-to-point
  hello-padding sometimes
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
  !
 !
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

IOS XE (R5):

```
!
interface Loopback0
 description Loopback
 ip address 12.39.5.5 255.255.255.255
 ip router isis core
 ipv6 address 2001:12:39:FF::5/128
 ipv6 router isis core
!
interface GigabitEthernet2
 description to SP2R3
 ip address 12.39.35.5 255.255.255.0
 ip router isis core
 negotiation auto
 ipv6 address FE80::5 link-local
 ipv6 address 2001:12:39:35::5/64
 ipv6 router isis core
 no mop enabled
 no mop sysid
 isis network point-to-point
!
interface GigabitEthernet3
 description to SP2R1
 ip address 12.39.15.5 255.255.255.0
 ip router isis core
 negotiation auto
 ipv6 address FE80::5 link-local
 ipv6 address 2001:12:39:15::5/64
 ipv6 router isis core
 no mop enabled
 no mop sysid
 isis network point-to-point
!
router isis core
 net 12.3900.0000.0000.0500
 is-type level-2-only
 router-id Loopback0
 metric-style wide
 !
 address-family ipv6
  multi-topology
  router-id Loopback0
 exit-address-family
!
```

Verify ISIS neighbors.

```
RP/0/0/CPU0:SP2R3#sh isis nei

IS-IS core neighbors:
System Id      Interface        SNPA           State Holdtime Type IETF-NSF
SP2R1          Gi0/0/0/6        *PtoP*         Up    23       L2   Capable
SP2R2          Gi0/0/0/0        *PtoP*         Up    29       L2   Capable
SP2R4          Gi0/0/0/2        *PtoP*         Up    28       L2   Capable
SP2-R5-RR      Gi0/0/0/3        *PtoP*         Up    23       L2   Capable

Total neighbor count: 4

SP2-R5-RR#sh isis nei

Tag core:
System Id       Type Interface     IP Address      State Holdtime Circuit Id
SP2R1           L2   Gi3           12.39.15.1      UP    25       00
SP2R3           L2   Gi2           12.39.35.3      UP    29       00
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Verify a topology and a database. There are two topologies one for IPv4 and one for IPv6 but just one database per router. In a multi-area ISIS topologies don't have to be congruent.

```
RP/0/0/CPU0:SP2R3#sh isis topology det

IS-IS core Level 2 IPv4 Unicast IS Topology
SP2R1 (12.39.1.1) [ucast 10 mcast <infinity>]
  via 12.39.13.1, GigabitEthernet0/0/0/6, SP2R1, Weight: 0
  parent SP2R3
  adv 0 critical, 0 high, 1 medium, 0 low priority prefixes
SP2R2 (12.39.2.2) [ucast 10 mcast <infinity>]
  via 12.39.23.2, GigabitEthernet0/0/0/0, SP2R2, Weight: 0
  parent SP2R3
  adv 0 critical, 0 high, 1 medium, 0 low priority prefixes
SP2R3 (12.39.3.3) [ucast 0 mcast <infinity>]
  child SP2R4
  child SP2R2
  child SP2R1
  child SP2-R5-RR
  adv 0 critical, 0 high, 0 medium, 0 low priority prefixes
SP2R4 (12.39.4.4) [ucast 10 mcast <infinity>]
  via 12.39.34.4, GigabitEthernet0/0/0/2, SP2R4, Weight: 0
  parent SP2R3
  adv 0 critical, 0 high, 1 medium, 0 low priority prefixes
SP2-R5-RR (12.39.5.5) [ucast 10 mcast <infinity>]
  via 12.39.35.5, GigabitEthernet0/0/0/3, SP2-R5-RR, Weight: 0
  parent SP2R3
  adv 0 critical, 0 high, 1 medium, 2 low priority prefixes

RP/0/0/CPU0:SP2R3#sh isis ipv6 topology det

IS-IS core Level 2 IPv6 Unicast IS Topology
SP2R1 (2001:12:39:ff::1) [ucast 10 mcast <infinity>]
  via fe80::1, GigabitEthernet0/0/0/6, SP2R1, Weight: 0
  parent SP2R3
  adv 0 critical, 0 high, 1 medium, 0 low priority prefixes
SP2R2 (2001:12:39:ff::2) [ucast 10 mcast <infinity>]
  via fe80::2, GigabitEthernet0/0/0/0, SP2R2, Weight: 0
  parent SP2R3
  adv 0 critical, 0 high, 1 medium, 0 low priority prefixes
SP2R3 (2001:12:39:ff::3) [ucast 0 mcast <infinity>]
  child SP2R4
  child SP2R2
  child SP2R1
  child SP2-R5-RR
  adv 0 critical, 0 high, 0 medium, 0 low priority prefixes
SP2R4 (2001:12:39:ff::4) [ucast 10 mcast <infinity>]
  via fe80::4, GigabitEthernet0/0/0/2, SP2R4, Weight: 0
  parent SP2R3
  adv 0 critical, 0 high, 1 medium, 0 low priority prefixes
SP2-R5-RR (2001:12:39:ff::5) [ucast 10 mcast <infinity>]
  via fe80::5, GigabitEthernet0/0/0/3, SP2-R5-RR, Weight: 0
  parent SP2R3
  adv 0 critical, 0 high, 1 medium, 2 low priority prefixes

RP/0/0/CPU0:SP2R3#sh isis data detail

IS-IS core (Level-2) Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
SP2R1.00-00           0x0000000c   0xa89a        1197            0/0/0
  Area Address:   1239
  NLPID:          0xcc
  NLPID:          0x8e
  MT:             Standard (IPv4 Unicast)
  MT:             IPv6 Unicast                                 0/0/0
  Hostname:       SP2R1
  IP Address:     12.39.1.1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  IPv6 Address:   2001:12:39:ff::1
  Metric: 10         IS-Extended SP2R2.00
  Metric: 10         IS-Extended SP2R4.00
  Metric: 10         IS-Extended SP2R3.00
  Metric: 10         IS-Extended SP2-R5-RR.00
  Metric: 0          IP-Extended 12.39.1.1/32
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R2.00
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R4.00
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R3.00
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2-R5-RR.00
  Metric: 0          MT (IPv6 Unicast) IPv6 2001:12:39:ff::1/128
SP2R2.00-00           0x0000000c   0x2e3c        1182            0/0/0
  Area Address:   1239
  NLPID:          0xcc
  NLPID:          0x8e
  MT:             Standard (IPv4 Unicast)
  MT:             IPv6 Unicast                                  0/0/0
  Hostname:       SP2R2
  IP Address:     12.39.2.2
  IPv6 Address:   2001:12:39:ff::2
  Metric: 10         IS-Extended SP2R1.00
  Metric: 10         IS-Extended SP2R4.00
  Metric: 10         IS-Extended SP2R3.00
  Metric: 0          IP-Extended 12.39.2.2/32
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R1.00
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R4.00
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R3.00
  Metric: 0          MT (IPv6 Unicast) IPv6 2001:12:39:ff::2/128
SP2R3.00-00         * 0x0000000e   0x75bf        521             0/0/0
  Area Address:   1239
  NLPID:          0xcc
  NLPID:          0x8e
  MT:             Standard (IPv4 Unicast)
  MT:             IPv6 Unicast                                  0/0/0
  Hostname:       SP2R3
  IP Address:     12.39.3.3
  IPv6 Address:   2001:12:39:ff::3
  Metric: 10         IS-Extended SP2R1.00
  Metric: 10         IS-Extended SP2R2.00
  Metric: 10         IS-Extended SP2R4.00
  Metric: 10         IS-Extended SP2-R5-RR.00
  Metric: 0          IP-Extended 12.39.3.3/32
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R1.00
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R2.00
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R4.00
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2-R5-RR.00
  Metric: 0          MT (IPv6 Unicast) IPv6 2001:12:39:ff::3/128
SP2R4.00-00           0x0000000b   0x76e8        1038            0/0/0
  Area Address:   1239
  NLPID:          0xcc
  NLPID:          0x8e
  MT:             Standard (IPv4 Unicast)
  MT:             IPv6 Unicast                                  0/0/0
  Hostname:       SP2R4
  IP Address:     12.39.4.4
  IPv6 Address:   2001:12:39:ff::4
  Metric: 10         IS-Extended SP2R1.00
  Metric: 10         IS-Extended SP2R2.00
  Metric: 10         IS-Extended SP2R3.00
  Metric: 0          IP-Extended 12.39.4.4/32
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R1.00
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R2.00
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R3.00
  Metric: 0          MT (IPv6 Unicast) IPv6 2001:12:39:ff::4/128
SP2-R5-RR.00-00       0x00000013   0x51bc        525             0/0/0
  Area Address:   1239
  NLPID:          0xcc
  NLPID:          0x8e
  MT:             Standard (IPv4 Unicast)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  MT:             IPv6 Unicast                                 0/0/0
  Router ID:      12.39.5.5
  Hostname:       SP2-R5-RR
  Metric: 10         IS-Extended SP2R3.00
  Metric: 10         IS-Extended SP2R1.00
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R1.00
  Metric: 10         MT (IPv6 Unicast) IS-Extended SP2R3.00
  IP Address:     12.39.5.5
  Metric: 10         IP-Extended 12.39.5.5/32
  Metric: 10         IP-Extended 12.39.15.0/24
  Metric: 10         IP-Extended 12.39.35.0/24
  IPv6 Address:   2001:12:39:ff::5
  IPv6 Router ID: 2001:12:39:ff::5
  Metric: 10         MT (IPv6 Unicast) IPv6 2001:12:39:15::/64
  Metric: 10         MT (IPv6 Unicast) IPv6 2001:12:39:35::/64
  Metric: 10         MT (IPv6 Unicast) IPv6 2001:12:39:ff::5/128

 Total Level-2 LSP count: 5     Local Level-2 LSP count: 1

SP2-R5-RR#sh isis topology

Tag core:

IS-IS TID 0 paths to level-2 routers
System Id            Metric     Next-Hop             Interface   SNPA
SP2R1                10         SP2R1                Gi3         fa16.3e96.1b84
SP2R2                20         SP2R1                Gi3         fa16.3e96.1b84
                                SP2R3                Gi2         fa16.3ed8.15a2
SP2R3                10         SP2R3                Gi2         fa16.3ed8.15a2
SP2R4                20         SP2R1                Gi3         fa16.3e96.1b84
                                SP2R3                Gi2         fa16.3ed8.15a2
SP2-R5-RR            --

SP2-R5-RR#sh isis ipv6 topology

Tag core:

IS-IS TID 2 paths to level-2 routers
System Id            Metric     Next-Hop             Interface   SNPA
SP2R1                10         SP2R1                Gi3         fa16.3e96.1b84
SP2R2                20         SP2R1                Gi3         fa16.3e96.1b84
                                SP2R3                Gi2         fa16.3ed8.15a2
SP2R3                10         SP2R3                Gi2         fa16.3ed8.15a2
SP2R4                20         SP2R1                Gi3         fa16.3e96.1b84
                                SP2R3                Gi2         fa16.3ed8.15a2
SP2-R5-RR#sh isis database detail

Tag core:
IS-IS Level-2 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime/Rcvd      ATT/P/OL
SP2R1.00-00           0x0000000C   0xA89A                879/1199      0/0/0
  Area Address: 12.39
  NLPID:        0xCC 0x8E
  Topology:     IPv4 (0x0)
                IPv6 (0x2)
  Hostname: SP2R1
  IP Address:   12.39.1.1
  IPv6 Address: 2001:12:39:FF::1
  Metric: 10         IS-Extended SP2R2.00
  Metric: 10         IS-Extended SP2R4.00
  Metric: 10         IS-Extended SP2R3.00
  Metric: 10         IS-Extended SP2-R5-RR.00
  Metric: 0          IP 12.39.1.1/32
  Metric: 10         IS (MT-IPv6) SP2R2.00
  Metric: 10         IS (MT-IPv6) SP2R4.00
  Metric: 10         IS (MT-IPv6) SP2R3.00
  Metric: 10         IS (MT-IPv6) SP2-R5-RR.00
  Metric: 0          IPv6 (MT-IPv6) 2001:12:39:FF::1/128
SP2R2.00-00           0x0000000C   0x2E3C                863/1198      0/0/0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Area Address: 12.39
  NLPID:        0xCC 0x8E
  Topology:     IPv4 (0x0)
                IPv6 (0x2)
  Hostname: SP2R2
  IP Address:   12.39.2.2
  IPv6 Address: 2001:12:39:FF::2
  Metric: 10         IS-Extended SP2R1.00
  Metric: 10         IS-Extended SP2R4.00
  Metric: 10         IS-Extended SP2R3.00
  Metric: 0          IP 12.39.2.2/32
  Metric: 10         IS (MT-IPv6) SP2R1.00
  Metric: 10         IS (MT-IPv6) SP2R4.00
  Metric: 10         IS (MT-IPv6) SP2R3.00
  Metric: 0          IPv6 (MT-IPv6) 2001:12:39:FF::2/128
SP2R3.00-00           0x0000000F   0x73C0                908/1199      0/0/0
  Area Address: 12.39
  NLPID:        0xCC 0x8E
  Topology:     IPv4 (0x0)
                IPv6 (0x2)
  Hostname: SP2R3
  IP Address:   12.39.3.3
  IPv6 Address: 2001:12:39:FF::3
  Metric: 10         IS-Extended SP2R1.00
  Metric: 10         IS-Extended SP2R2.00
  Metric: 10         IS-Extended SP2R4.00
  Metric: 10         IS-Extended SP2-R5-RR.00
  Metric: 0          IP 12.39.3.3/32
  Metric: 10         IS (MT-IPv6) SP2R1.00
  Metric: 10         IS (MT-IPv6) SP2R2.00
  Metric: 10         IS (MT-IPv6) SP2R4.00
  Metric: 10         IS (MT-IPv6) SP2-R5-RR.00
  Metric: 0          IPv6 (MT-IPv6) 2001:12:39:FF::3/128
SP2R4.00-00           0x0000000B   0x76E8                719/1198      0/0/0
  Area Address: 12.39
  NLPID:        0xCC 0x8E
  Topology:     IPv4 (0x0)
                IPv6 (0x2)
  Hostname: SP2R4
  IP Address:   12.39.4.4
  IPv6 Address: 2001:12:39:FF::4
  Metric: 10         IS-Extended SP2R1.00
  Metric: 10         IS-Extended SP2R2.00
  Metric: 10         IS-Extended SP2R3.00
  Metric: 0          IP 12.39.4.4/32
  Metric: 10         IS (MT-IPv6) SP2R1.00
  Metric: 10         IS (MT-IPv6) SP2R2.00
  Metric: 10         IS (MT-IPv6) SP2R3.00
  Metric: 0          IPv6 (MT-IPv6) 2001:12:39:FF::4/128
SP2-R5-RR.00-00     * 0x00000014   0x4FBD                997/*         0/0/0
  Area Address: 12.39
  NLPID:        0xCC 0x8E
  Topology:     IPv4 (0x0)
                IPv6 (0x2)
  Router ID:    12.39.5.5
  Hostname: SP2-R5-RR
  Metric: 10         IS-Extended SP2R3.00
  Metric: 10         IS-Extended SP2R1.00
  Metric: 10         IS (MT-IPv6) SP2R1.00
  Metric: 10         IS (MT-IPv6) SP2R3.00
  IP Address:   12.39.5.5
  Metric: 10         IP 12.39.5.5/32
  Metric: 10         IP 12.39.15.0/24
  Metric: 10         IP 12.39.35.0/24
  IPv6 Address: 2001:12:39:FF::5
  IPv6 Router ID: 2001:12:39:FF::5
  Metric: 10         IPv6 (MT-IPv6) 2001:12:39:15::/64
  Metric: 10         IPv6 (MT-IPv6) 2001:12:39:35::/64
  Metric: 10         IPv6 (MT-IPv6) 2001:12:39:FF::5/128
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Verification of the routing table. Check all loopbacks.

```
RP/0/0/CPU0:SP2R3#sh route isis

i L2 12.39.1.1/32 [115/10] via 12.39.13.1, 01:03:11, GigabitEthernet0/0/0/6
i L2 12.39.2.2/32 [115/10] via 12.39.23.2, 01:03:26, GigabitEthernet0/0/0/0
i L2 12.39.4.4/32 [115/10] via 12.39.34.4, 01:04:29, GigabitEthernet0/0/0/2
i L2 12.39.5.5/32 [115/20] via 12.39.35.5, 00:26:42, GigabitEthernet0/0/0/3
i L2 12.39.15.0/24 [115/20] via 12.39.35.5, 00:18:24, GigabitEthernet0/0/0/3

RP/0/0/CPU0:SP2R3#sh route ipv6 isis

i L2 2001:12:39:15::/64
      [115/20] via fe80::5, 00:17:08, GigabitEthernet0/0/0/3
i L2 2001:12:39:ff::1/128
      [115/10] via fe80::1, 01:02:52, GigabitEthernet0/0/0/6
i L2 2001:12:39:ff::2/128
      [115/10] via fe80::2, 01:03:07, GigabitEthernet0/0/0/0
i L2 2001:12:39:ff::4/128
      [115/10] via fe80::4, 01:04:10, GigabitEthernet0/0/0/2
i L2 2001:12:39:ff::5/128
      [115/20] via fe80::5, 00:17:08, GigabitEthernet0/0/0/3

SP2-R5-RR#sh ip route isis
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      12.0.0.0/8 is variably subnetted, 9 subnets, 2 masks
i L2     12.39.1.1/32 [115/10] via 12.39.15.1, 00:52:22, GigabitEthernet3
i L2     12.39.2.2/32 [115/20] via 12.39.35.3, 00:52:22, GigabitEthernet2
                      [115/20] via 12.39.15.1, 00:52:22, GigabitEthernet3
i L2     12.39.3.3/32 [115/10] via 12.39.35.3, 00:52:40, GigabitEthernet2
i L2     12.39.4.4/32 [115/20] via 12.39.35.3, 00:52:22, GigabitEthernet2
                      [115/20] via 12.39.15.1, 00:52:22, GigabitEthernet3

SP2-R5-RR#sh ipv6 route isis
IPv6 Routing Table - default - 10 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
       NDr - Redirect, RL - RPL, O - OSPF Intra, OI - OSPF Inter
       OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1
       ON2 - OSPF NSSA ext 2, la - LISP alt, lr - LISP site-registrations
       ld - LISP dyn-eid, lA - LISP away, a - Application
I2  2001:12:39:FF::1/128 [115/10]
     via FE80::1, GigabitEthernet3
I2  2001:12:39:FF::2/128 [115/20]
     via FE80::1, GigabitEthernet3
     via FE80::3, GigabitEthernet2
I2  2001:12:39:FF::3/128 [115/10]
     via FE80::3, GigabitEthernet2
I2  2001:12:39:FF::4/128 [115/20]
     via FE80::1, GigabitEthernet3
     via FE80::3, GigabitEthernet2
```

Traffic flow verification.

```
SP2-R5-RR#ping 12.39.4.4 source 12.39.5.5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.39.4.4, timeout is 2 seconds:
Packet sent with a source address of 12.39.5.5
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/13/54 ms

SP2-R5-RR#ping 2001:12:39:FF::2 source 2001:12:39:FF::5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:12:39:FF::2, timeout is 2 seconds:
Packet sent with a source address of 2001:12:39:FF::5
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/11/19 ms
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Secure the ISIS domain

Turn on the Hello and LSP authentication in the Level-2 area of ISIS.

IOS XR (R3):

```
RP/0/0/CPU0:SP2R3#sh run key chain
key chain isis-keys
 key 1
  accept-lifetime 00:00:00 january 01 2017 infinite
  key-string password 121A0C041104
  send-lifetime 00:00:00 january 01 2017 infinite
  cryptographic-algorithm HMAC-MD5
 !
 accept-tolerance infinite
!

SP1-R5-RR#sh run | section key chain
key chain isis-keys
 key 1
  key-string cisco
```

IOS XR (R3):

```
router isis core
 is-type level-2-only
 net 1239.0000.0000.0003.00
 nsr
 lsp-password keychain isis-keys
 address-family ipv4 unicast
  metric-style wide
  advertise passive-only
 !
 address-family ipv6 unicast
  advertise passive-only
 !
 interface Loopback0
  passive
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
  !
 !
 interface GigabitEthernet0/0/0/0
  mesh-group 1239
  point-to-point
  hello-padding sometimes
  hello-password keychain isis-keys
  address-family ipv4 unicast
   fast-reroute per-prefix
  !
  address-family ipv6 unicast
  !
 !
 interface GigabitEthernet0/0/0/2
  mesh-group 1239
  point-to-point
  hello-padding sometimes
  hello-password keychain isis-keys
  address-family ipv4 unicast
   fast-reroute per-prefix
  !
  address-family ipv6 unicast
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  !
 !
 interface GigabitEthernet0/0/0/3
  point-to-point
  hello-padding sometimes
  hello-password keychain isis-keys
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
  !
 !
 interface GigabitEthernet0/0/0/6
  mesh-group 1239
  point-to-point
  hello-padding sometimes
  hello-password keychain isis-keys
  address-family ipv4 unicast
   fast-reroute per-prefix
  !
  address-family ipv6 unicast
  !
 !
```

IOS XE (R5):

```
!
interface GigabitEthernet2
 description to SP2R3
 ip address 12.39.35.5 255.255.255.0
 ip router isis core
 negotiation auto
 ipv6 address FE80::5 link-local
 ipv6 address 2001:12:39:35::5/64
 ipv6 router isis core
 no mop enabled
 no mop sysid
 isis network point-to-point
 isis authentication mode md5
 isis authentication key-chain isis-keys
!
interface GigabitEthernet3
 description to SP2R1
 ip address 12.39.15.5 255.255.255.0
 ip router isis core
 negotiation auto
 ipv6 address FE80::5 link-local
 ipv6 address 2001:12:39:15::5/64
 ipv6 router isis core
 no mop enabled
 no mop sysid
 isis network point-to-point
 isis authentication mode md5
 isis authentication key-chain isis-keys
!
router isis core
 net 12.3900.0000.0000.0500
 is-type level-2-only
 router-id Loopback0
 authentication mode md5
 authentication key-chain isis-keys
 metric-style wide
 !
 address-family ipv6
  multi-topology
  router-id Loopback0
 exit-address-family
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Verification of ISIS sessions and prefix advertisement.

```
RP/0/0/CPU0:SP2R3#sh isis nei

IS-IS core neighbors:
System Id      Interface        SNPA           State Holdtime Type IETF-NSF
SP2R1          Gi0/0/0/6        *PtoP*         Up    27       L2   Capable
SP2R2          Gi0/0/0/0        *PtoP*         Up    21       L2   Capable
SP2R4          Gi0/0/0/2        *PtoP*         Up    24       L2   Capable
SP2-R5-RR      Gi0/0/0/3        *PtoP*         Up    27       L2   Capable
Total neighbor count: 4

RP/0/0/CPU0:SP2R3#sh route isis

i L2 12.39.1.1/32 [115/10] via 12.39.13.1, 00:12:50, GigabitEthernet0/0/0/6
                  [115/20] via 12.39.23.2, 00:12:50, GigabitEthernet0/0/0/0 (!)
i L2 12.39.2.2/32 [115/20] via 12.39.13.1, 00:12:50, GigabitEthernet0/0/0/6 (!)
                  [115/10] via 12.39.23.2, 00:12:50, GigabitEthernet0/0/0/0
i L2 12.39.4.4/32 [115/10] via 12.39.34.4, 00:12:53, GigabitEthernet0/0/0/2
                  [115/20] via 12.39.23.2, 00:12:53, GigabitEthernet0/0/0/0 (!)
i L2 12.39.5.5/32 [115/20] via 12.39.35.5, 00:12:50, GigabitEthernet0/0/0/3
i L2 12.39.15.0/24 [115/20] via 12.39.35.5, 00:12:50, GigabitEthernet0/0/0/3

RP/0/0/CPU0:SP2R3#sh isis dat det SP2-R5-RR.00-00

IS-IS core (Level-1) Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
The requested LSP SP2-R5-RR.00-00 was not found in the IS-IS core Level-1 LSP Database

IS-IS core (Level-2) Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
SP2-R5-RR.00-00       0x00000088   0xd576        752             0/0/0
  Auth:             Algorithm HMAC-MD5, Length: 17
  Area Address:     1239
  NLPID:            0xcc
  NLPID:            0x8e
  MT:               Standard (IPv4 Unicast)
  MT:               IPv6 Unicast                                     0/0/0
  Router ID:        12.39.5.5
  Hostname:         SP2-R5-RR
  Metric: 10          IS-Extended SP2R1.00
  Metric: 10          IS-Extended SP2R3.00
  Metric: 10          MT (IPv6 Unicast) IS-Extended SP2R1.00
  Metric: 10          MT (IPv6 Unicast) IS-Extended SP2R3.00
  IP Address:       12.39.5.5
  Metric: 10          IP-Extended 12.39.5.5/32
  Metric: 10          IP-Extended 12.39.15.0/24
  Metric: 10          IP-Extended 12.39.35.0/24
  Metric: 10          IP-Extended 12.39.1.1/32
  IPv6 Address:     2001:12:39:ff::5
  IPv6 Router ID: 2001:12:39:ff::5
  Metric: 10          MT (IPv6 Unicast) IPv6 2001:12:39:15::/64
  Metric: 10          MT (IPv6 Unicast) IPv6 2001:12:39:35::/64
  Metric: 10          MT (IPv6 Unicast) IPv6 2001:12:39:ff::5/128

RP/0/0/CPU0:SP2R3#sh isis trace detailed | utility egrep 'AUTH' | utility tail count 10
Jan 13 00:56:11.291 isis/core/det 0/0/CPU0 t8  UPD_CSNP_SEND_ADD_AUTH L2 Gi0/0/0/2
Jan 13 00:56:11.301 isis/core/std 0/0/CPU0 t8  LOCAL_SYSTEM_ADD_AUTH L2 Node Num: 0
Jan 13 00:56:11.301 isis/core/det 0/0/CPU0 t8  UPD_LSP_UPDATE_AUTH Upd Type: LSP L2
0000.0000.0003.00-00
Jan 13 00:56:11.511 isis/core/std 0/0/CPU0 t8  LOCAL_SYSTEM_ADD_AUTH L2 Node Num: 0
Jan 13 00:56:11.541 isis/core/det 0/0/CPU0 t8  UPD_CSNP_SEND_ADD_AUTH L2 Gi0/0/0/0
Jan 13 00:56:12.291 isis/core/det 0/0/CPU0 t8  UPD_PSNP_SEND_ADD_AUTH L2 Gi0/0/0/2
Jan 13 00:56:12.611 isis/core/std 0/0/CPU0 t8  LOCAL_SYSTEM_ADD_AUTH L2 Node Num: 0
Jan 13 00:56:13.051 isis/core/det 0/0/CPU0 t8  UPD_CSNP_SEND_ADD_AUTH L2 Gi0/0/0/3
Jan 13 00:56:14.071 isis/core/det 0/0/CPU0 t8  UPD_PSNP_SEND_ADD_AUTH L2 Gi0/0/0/6
Jan 13 01:07:37.654 isis/core/det 0/0/CPU0 t8  UPD_PSNP_SEND_ADD_AUTH L2 Gi0/0/0/6
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

AS1239 IP LFA

Deploy IP LFA configuration in AS1239.

Task 1: AS1239 IP LFA

Enable IP LFA (IP Fast Reroute) which is a make-before-break solution for the IP network. IP LFA is local and autonomously calculated on each of the routers so you can configure one or more devices even if some routers do not support it.

Configuration example.

IOS XR (R3)

```
router isis core
 !
 interface GigabitEthernet0/0/0/0
  address-family ipv4 unicast
   fast-reroute per-prefix
  !
  address-family ipv6 unicast
  !
 !
 interface GigabitEthernet0/0/0/2
  address-family ipv4 unicast
   fast-reroute per-prefix
  !
  address-family ipv6 unicast
  !
 !
 interface GigabitEthernet0/0/0/6
  mesh-group 1239
  point-to-point
  hello-padding sometimes
  address-family ipv4 unicast
   fast-reroute per-prefix
  !
  address-family ipv6 unicast
  !
 !
!
```

Verification of protected prefixes.

```
RP/0/0/CPU0:SP2R3#sh isis fast-reroute summary

IS-IS core IPv4 Unicast FRR summary

                          Critical   High       Medium     Low        Total
                          Priority   Priority   Priority   Priority
Prefixes reachable in L2
  All paths protected     0          0          3          0          3
  Some paths protected    0          0          0          0          0
  Unprotected             0          0          1          2          3
  Protection coverage     0.00%      0.00%      75.00%     0.00%      50.00%
```

Verification of RIB and FIB entries.

```
RP/0/0/CPU0:SP2R3#sh route
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Wed Jan 12 03:36:05.583 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

Gateway of last resort is not set

C    10.0.128.0/29 is directly connected, 2d04h, GigabitEthernet0/0/0/4
L    10.0.128.3/32 is directly connected, 2d04h, GigabitEthernet0/0/0/4
i L2 12.39.1.1/32 [115/20] via 12.39.23.2, 00:04:12, GigabitEthernet0/0/0/0 (!)
                  [115/10] via 12.39.13.1, 00:04:12, GigabitEthernet0/0/0/6
i L2 12.39.2.2/32 [115/10] via 12.39.23.2, 00:04:12, GigabitEthernet0/0/0/0
                  [115/20] via 12.39.13.1, 00:04:12, GigabitEthernet0/0/0/6 (!)
L    12.39.3.3/32 is directly connected, 2d04h, Loopback0
i L2 12.39.4.4/32 [115/20] via 12.39.23.2, 00:04:12, GigabitEthernet0/0/0/0 (!)
                  [115/10] via 12.39.34.4, 00:04:12, GigabitEthernet0/0/0/2
i L2 12.39.5.5/32 [115/20] via 12.39.35.5, 01:14:58, GigabitEthernet0/0/0/3
C    12.39.13.0/24 is directly connected, 02:00:19, GigabitEthernet0/0/0/6
L    12.39.13.3/32 is directly connected, 02:00:19, GigabitEthernet0/0/0/6
i L2 12.39.15.0/24 [115/20] via 12.39.35.5, 01:06:41, GigabitEthernet0/0/0/3
C    12.39.23.0/24 is directly connected, 2d04h, GigabitEthernet0/0/0/0
L    12.39.23.3/32 is directly connected, 2d04h, GigabitEthernet0/0/0/0
C    12.39.34.0/24 is directly connected, 2d04h, GigabitEthernet0/0/0/2
L    12.39.34.3/32 is directly connected, 2d04h, GigabitEthernet0/0/0/2
C    12.39.35.0/24 is directly connected, 2d04h, GigabitEthernet0/0/0/3
L    12.39.35.3/32 is directly connected, 2d04h, GigabitEthernet0/0/0/3
C    22.22.33.0/24 is directly connected, 2d04h, GigabitEthernet0/0/0/1
L    22.22.33.254/32 is directly connected, 2d04h, GigabitEthernet0/0/0/1
C    22.42.37.0/24 is directly connected, 2d04h, GigabitEthernet0/0/0/5
L    22.42.37.3/32 is directly connected, 2d04h, GigabitEthernet0/0/0/5


RP/0/0/CPU0:SP2R3#sh cef fast-reroute

Prefix              Next Hop            Interface
------------------- ------------------- ------------------
12.39.1.1/32        12.39.23.2/32       GigabitEthernet0/0/0/0 (!)
                    12.39.13.1/32       GigabitEthernet0/0/0/6
12.39.2.2/32        12.39.23.2/32       GigabitEthernet0/0/0/0
                    12.39.13.1/32       GigabitEthernet0/0/0/6 (!)
12.39.4.4/32        12.39.23.2/32       GigabitEthernet0/0/0/0 (!)
                    12.39.34.4/32       GigabitEthernet0/0/0/2

RP/0/0/CPU0:SP2R3#sh cef fast-reroute det
Wed Jan 12 03:32:25.788 UTC
12.39.1.1/32, version 38, internal 0x1000001 0x0 (ptr 0xa149b574) [1], 0x0 (0xa1465b90),
0x0 (0x0)
 Updated Jan 12 03:31:53.300
 local adjacency 12.39.13.1
 Prefix Len 32, traffic index 0, precedence n/a, priority 1
   via 12.39.23.2/32, GigabitEthernet0/0/0/0, 8 dependencies, weight 0, class 0, backup
[flags 0x300]
    path-idx 0 NHID 0x0 [0xa0fb5694 0x0]
    next hop 12.39.23.2/32
    local adjacency
   via 12.39.13.1/32, GigabitEthernet0/0/0/6, 8 dependencies, weight 0, class 0,
protected [flags 0x400]
    path-idx 1 bkup-idx 0 NHID 0x0 [0xa0ce863c 0x0]
    next hop 12.39.13.1/32
12.39.2.2/32, version 40, internal 0x1000001 0x0 (ptr 0xa149abf4) [1], 0x0 (0xa14658e4),
0x0 (0x0)
 Updated Jan 12 03:31:53.300
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
 local adjacency 12.39.23.2
 Prefix Len 32, traffic index 0, precedence n/a, priority 1
   via 12.39.23.2/32, GigabitEthernet0/0/0/0, 6 dependencies, weight 0, class 0,
protected [flags 0x400]
    path-idx 0 bkup-idx 1 NHID 0x0 [0xa0ce880c 0x0]
    next hop 12.39.23.2/32
   via 12.39.13.1/32, GigabitEthernet0/0/0/6, 6 dependencies, weight 0, class 0, backup
[flags 0x300]
    path-idx 1 NHID 0x0 [0xa0fb53f4 0x0]
    next hop 12.39.13.1/32
    local adjacency
12.39.4.4/32, version 39, internal 0x1000001 0x0 (ptr 0xa149ad74) [1], 0x0 (0xa1465950),
0x0 (0x0)
 Updated Jan 12 03:31:53.300
 local adjacency 12.39.34.4
 Prefix Len 32, traffic index 0, precedence n/a, priority 1
   via 12.39.23.2/32, GigabitEthernet0/0/0/0, 8 dependencies, weight 0, class 0, backup
[flags 0x300]
    path-idx 0 NHID 0x0 [0xa0fb5694 0x0]
    next hop 12.39.23.2/32
    local adjacency
   via 12.39.34.4/32, GigabitEthernet0/0/0/2, 8 dependencies, weight 0, class 0,
protected [flags 0x400]
    path-idx 1 bkup-idx 0 NHID 0x0 [0xa0ce8724 0x0]
    next hop 12.39.34.4/32
```

ISIS verification of calculated paths.

```
RP/0/0/CPU0:SP2R3#sh isis fast-reroute

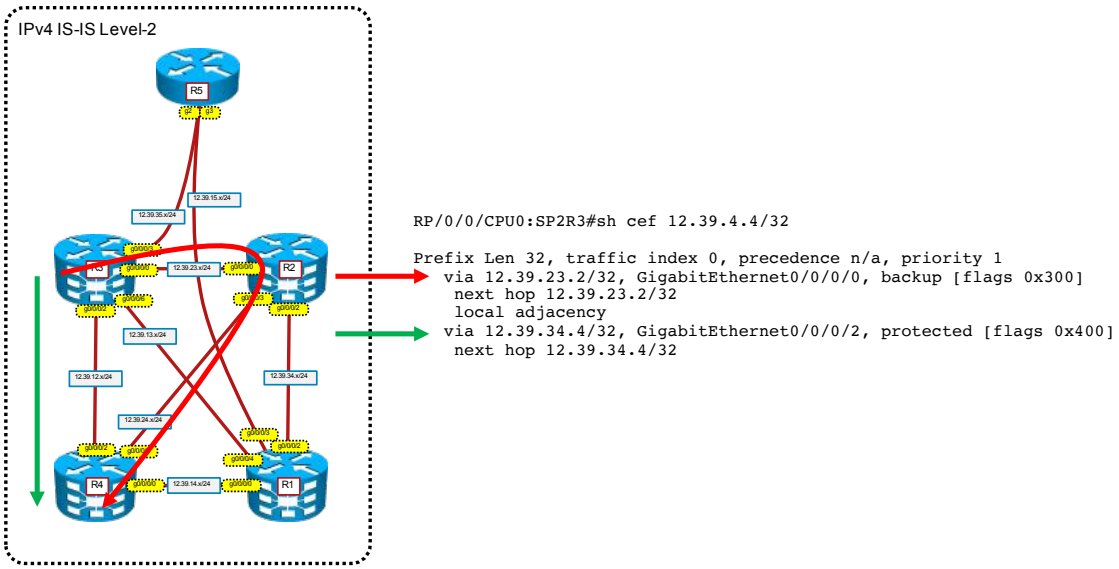
IS-IS core IPv4 Unicast FRR backups

Codes: L1 - level 1, L2 - level 2, ia - interarea (leaked into level 1)
       df - level 1 default (closest attached router), su - summary null
       C - connected, S - static, R - RIP, B - BGP, O - OSPF
       E - EIGRP, A - access/subscriber, M - mobile, a - application
       i - IS-IS (redistributed from another instance)
       D - Downstream, LC - Line card disjoint, NP - Node protecting
       P - Primary path, SRLG - SRLG disjoint, TM - Total metric via backup

Maximum parallel path count: 8

L2 12.39.1.1/32 [10/115]
     via 12.39.13.1, GigabitEthernet0/0/0/6, SP2R1, Weight: 0
       FRR backup via 12.39.23.2, GigabitEthernet0/0/0/0, SP2R2, Weight: 0, Metric: 20
L2 12.39.2.2/32 [10/115]
     via 12.39.23.2, GigabitEthernet0/0/0/0, SP2R2, Weight: 0
       FRR backup via 12.39.13.1, GigabitEthernet0/0/0/6, SP2R1, Weight: 0, Metric: 20
L2 12.39.4.4/32 [10/115]
     via 12.39.34.4, GigabitEthernet0/0/0/2, SP2R4, Weight: 0
       FRR backup via 12.39.23.2, GigabitEthernet0/0/0/0, SP2R2, Weight: 0, Metric: 20
L2 12.39.5.5/32 [20/115]
     via 12.39.35.5, GigabitEthernet0/0/0/3, SP2-R5-RR, Weight: 0
       No FRR backup
L2 12.39.15.0/24 [20/115]
     via 12.39.35.5, GigabitEthernet0/0/0/3, SP2-R5-RR, Weight: 0
       No FRR backup
L2 12.39.35.0/24 [20/115]
     via 12.39.35.5, GigabitEthernet0/0/0/3, SP2-R5-RR, Weight: 0
       No FRR backup
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński




```
RP/0/0/CPU0:SP2R3#sh isis fast-reroute 12.39.4.4/32

L2 12.39.4.4/32 [10/115]
     via 12.39.34.4, GigabitEthernet0/0/0/2, SP2R4, Weight: 0
       FRR backup via 12.39.23.2, GigabitEthernet0/0/0/0, SP2R2, Weight: 0, Metric: 20

RP/0/0/CPU0:SP2R3#sh cef 12.39.4.4/32

12.39.4.4/32, version 39, internal 0x1000001 0x0 (ptr 0xa149ad74) [1], 0x0 (0xa1465950),
0x0 (0x0)
 Updated Jan 12 03:31:53.300
 local adjacency 12.39.34.4
 Prefix Len 32, traffic index 0, precedence n/a, priority 1
   via 12.39.23.2/32, GigabitEthernet0/0/0/0, 8 dependencies, weight 0, class 0, backup
[flags 0x300]
    path-idx 0 NHID 0x0 [0xa0fb5694 0x0]
    next hop 12.39.23.2/32
    local adjacency
   via 12.39.34.4/32, GigabitEthernet0/0/0/2, 8 dependencies, weight 0, class 0,
protected [flags 0x400]
    path-idx 1 bkup-idx 0 NHID 0x0 [0xa0ce8724 0x0]
    next hop 12.39.34.4/32

    Load distribution: 0 (refcount 2)

    Hash  OK  Interface                 Address
    0     Y   GigabitEthernet0/0/0/2    12.39.34.4
```

How to configure and verify ISIS IP LFA on IOS XE?

Configuration:

```
SP2-R5-RR(config)#router isis AS1239
SP2-R5-RR(config-router)#fast-reroute per-prefix level-2 all

SP2-R5-RR#sh isis fast-reroute summ
Tag AS1239:
Microloop Avoidance State: Disabled
IPv4 Fast-Reroute Protection Summary:

 Prefix Counts:          Total      Protected    Coverage
   High priority:        0          0            0%
   Normal priority:      12         10           83%
   Total:                12         10           83%
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Verification: there is no clear indication of IP LFA in this case as you can see just one entry for 12.39.3.3/32. No IP LFA marks for other prefixes.

```
SP2SP2-R5-RR#sh ip route

SP2-R5-RR#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      12.0.0.0/8 is variably subnetted, 16 subnets, 3 masks
B        12.39.0.0/16 [200/0] via 12.39.1.1, 17:07:40
i L2     12.39.1.1/32 [115/20] via 12.39.15.1, 00:03:50, GigabitEthernet3
i L2     12.39.2.2/32 [115/30] via 12.39.35.3, 00:03:50, GigabitEthernet2
                      [115/30] via 12.39.15.1, 00:03:50, GigabitEthernet3
i L2     12.39.3.3/32 [115/20] via 12.39.35.3, 00:03:50, GigabitEthernet2
i L2     12.39.4.4/32 [115/30] via 12.39.35.3, 00:03:50, GigabitEthernet2
                      [115/30] via 12.39.15.1, 00:03:50, GigabitEthernet3
```

With more detailed commands a repair path is visible.

```
SP2-R5-RR#sh ip route repair-paths
[..]
Gateway of last resort is not set

      12.0.0.0/8 is variably subnetted, 16 subnets, 3 masks
B        12.39.0.0/16 [200/0] via 12.39.1.1, 17:11:05
i L2     12.39.1.1/32 [115/20] via 12.39.15.1, 00:07:15, GigabitEthernet3
                              Repair Path: 12.39.35.3, via GigabitEthernet2
                      [RPR][115/30] via 12.39.35.3, 00:07:15, GigabitEthernet2
i L2     12.39.2.2/32 [115/30] via 12.39.35.3, 00:07:15, GigabitEthernet2
                              Repair Path: 12.39.15.1, via GigabitEthernet3
                      [115/30] via 12.39.15.1, 00:07:15, GigabitEthernet3
                              Repair Path: 12.39.35.3, via GigabitEthernet2
i L2     12.39.3.3/32 [115/20] via 12.39.35.3, 00:07:15, GigabitEthernet2
                              Repair Path: 12.39.15.1, via GigabitEthernet3
                      [RPR][115/30] via 12.39.15.1, 00:07:15, GigabitEthernet3
i L2     12.39.4.4/32 [115/30] via 12.39.35.3, 00:07:19, GigabitEthernet2
                              Repair Path: 12.39.15.1, via GigabitEthernet3

SP2-R5-RR#sh ip route 12.39.3.3 255.255.255.255
Routing entry for 12.39.3.3/32
  Known via "isis", distance 115, metric 20, type level-2
  Redistributing via isis AS1239
  Last update from 12.39.35.3 on GigabitEthernet2, 00:02:46 ago
  Routing Descriptor Blocks:
  * 12.39.35.3, from 12.39.3.3, 00:02:46 ago, via GigabitEthernet2
      Route metric is 20, traffic share count is 1
      Repair Path: 12.39.15.1, via GigabitEthernet3

SP2-R5-RR#sh ip cef 12.39.3.3 det
12.39.3.3/32, epoch 2
  dflt local label info: global/123503 [0x3]
  1 RR source [non-eos indirection, heavily shared]
  nexthop 12.39.35.3 GigabitEthernet2 label [explicit-null|123103]-(local:123503)
    repair: attached-nexthop 12.39.15.1 GigabitEthernet3
  nexthop 12.39.15.1 GigabitEthernet3, repair
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 4: Basic BGP

For the specific sections we will use final IGP topology built during previous session, but some AS configuration may change, or require from you reconfiguration. Please pay attention to topology drawings and reconfiguration tasks.

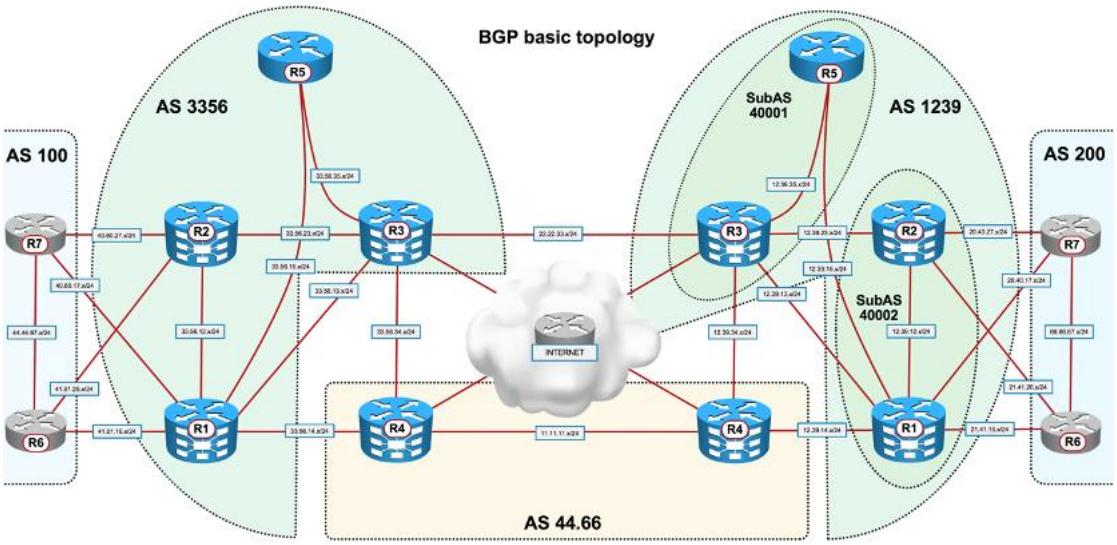
This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch04.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch04.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch04.unl`

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: Basic BGP configuration

In tasks 1 to 4 we will build and experiment with below logical topology:





During tasks 1-4, we'll experiment with the BGP configuration options and IOS-XE/XR differences.

While it's fair call to expect on CCIE SP practical lab exam any obscure configuration option to be tested, bear in mind they may not be best practices.

In task 5 we'll build a whole new network from the scratch. There, most of real world best-practices will be used. Based on this topology we will configure some advanced features in session 4.

First task will be to configure and peer routers in AS 100, AS 3356, AS 44.66, AS 1239 and AS200.

Task 1.1: BGP in AS 100

Task 1.1.1: Simple iBGP session

Configure IOS-XE routers R7 and R6 to peer over physical links first:

```
CEL6(config)#router bgp 100
CEL6(config-router)#neighbor 44.44.67.7 remote-as 100
CEL6(config-router)#^Z
CEL6#

CEL7(config)#router bgp 100
CEL7(config-router)#neighbor 44.44.67.6 remote-as 100
CEL7(config-router)#^Z
CEL7#
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

If the routers are correctly connected, the session should come up almost immediately:

```
CEL6#
*20:27:14.250: %SYS-5-CONFIG_I: Configured from console by console
*20:27:42.155: %BGP-5-ADJCHANGE: neighbor 44.44.67.7 Up

CEL7#
*20:27:15.969: %SYS-5-CONFIG_I: Configured from console by console
*20:27:42.565: %BGP-5-ADJCHANGE: neighbor 44.44.67.6 Up
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.1.2: BGP router-id selection



By default, IOS and IOS-XE select the highest-numbered loopback interface as BGP router-id. If they don't exist, highest IP address from physical interfaces will be allocated. When reconfiguring the router-id, all existing BGP sessions will reset.

It is recommended to configure Loopback within own IP address space per each router solely for the purpose of assigning stable router-id's. Properly executed it has a benefit of being able to identify the routers at the same time.

Router CEL6 has one IP on Loopback0: 172.16.6.6, however CEL7 has two Loopback interfaces. Loopback1 is effect of old configuration and as it has higher IP address of two, it will be used as router-id:

```
CEL7#sh ip int brief | in Loo
Loopback0              172.16.7.7      YES TFTP   up                    up
Loopback1              172.16.77.77    YES TFTP   up                    up
```

This makes BGP pick up 172.16.77.77 as router-id which may not be optimal:

```
CEL7#sh ip bgp summary | i router
BGP router identifier 172.16.77.77, local AS number 100
```

We'll force the change of router-id for CEL7 – this will cause all sessions to reset:

```
CEL7(config)#router bgp 100
CEL7(config-router)#bgp router-id interface loopback 0
*07:59:36.268: %BGP-5-ADJCHANGE: neighbor 44.44.67.6 Down Router ID changed
*07:59:36.799: %BGP-5-ADJCHANGE: neighbor 44.44.67.6 Up
```

Now the CEL7 has 'proper' router-id from our address plan:

```
CEL7#sh ip bgp summary | i router
BGP router identifier 172.16.7.7, local AS number 100
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.1.3: Advertise network in BGP using ‘network’ statement

Please note, because we didn’t configure any networks for BGP to advertise, nothing is advertised from both nodes:

```
CEL6#sh ip bgp neighbors 44.44.67.7 advertised-routes

Total number of prefixes 0
CEL7#sh ip bgp neighbors 44.44.67.6 advertised-routes

Total number of prefixes 0
```

We can also check summary of the BGP state on the IOS-XE devices:

```
CEL6#sh ip bgp summary
BGP router identifier 172.16.6.6, local AS number 100
BGP table version is 1, main routing table version 1

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
44.44.67.7      4          100       7       7        1    0    0 00:03:18        0
```

As for the neighbor status, we can see that we have some **additional capabilities announced and received**, we’re using by default **Path MTU Discovery** (with TCP MSS properly negotiated to 1460 bytes), and that the **source port is 21350/tcp to remote 179/tcp** (for now, output is truncated to focus on most important information):

```
CEL6#sh ip bgp neighbors
BGP neighbor is 44.44.67.7,  remote AS 100, internal link
  BGP version 4, remote router ID 172.16.77.77
  BGP state = Established, up for 00:03:20
  Last read 00:00:46, last write 00:00:38, hold time is 180, keepalive interval is 60
seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
[…]

  Interface associated: (none) (peering address in same link)
[…]
Local host: 44.44.67.6, Local port: 21350
Foreign host: 44.44.67.7, Foreign port: 179
[…]
Datagrams (max data segment is 1460 bytes):
Rcvd: 14 (out of order: 0), with data: 7, total data bytes: 175
Sent: 13 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion: 0), with
data: 7, total data bytes: 175
```

For test, we’ll advertise the connected AS100<>AS3356 links from both routers using the easiest way:

```
CEL6(config)#router bgp 100
CEL6(config-router)#network 41.61.16.0 mask 255.255.255.0

CEL7(config)#router bgp 100
CEL7(config-router)#network 40.60.27.0 mask 255.255.255.0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

You should see the network advertised from CEL6 to CEL7:

```
CEL6#sh ip bgp neighbors 44.44.67.7 advertised-routes
[…]
     Network          Next Hop            Metric LocPrf Weight Path
 *>   41.61.16.0/24    0.0.0.0                  0         32768 i
```

…as well, as from CEL7 to CEL6:

```
CEL7#sh ip bgp neighbors 44.44.67.6 advertised-routes
[…]
     Network          Next Hop            Metric LocPrf Weight Path
 *>   40.60.27.0/24    0.0.0.0                  0         32768 i

Total number of prefixes 1
```

We can check routers BGP table – for CEL6:

```
CEL6#sh ip bgp
BGP table version is 5, local router ID is 172.16.6.6
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i  40.60.27.0/24    44.44.67.7               0    100      0 i
 *>   41.61.16.0/24    0.0.0.0                  0         32768 i
```

Next Hop of 0.0.0.0 means this node is originating the route. Note, that for the prefix advertised by remote router over iBGP session, the next hop is address on the link. Likewise, for CEL7:

```
CEL7#sh ip bgp
BGP table version is 5, local router ID is 172.16.77.77
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>   40.60.27.0/24    0.0.0.0                  0         32768 i
 *>i  41.61.16.0/24    44.44.67.6               0    100      0 i
```

The BGP entry for this remote prefix can be viewed on IOS & IOS-XE by executing:

```
CEL6#sh ip bgp 40.60.27.0
BGP routing table entry for 40.60.27.0/24, version 5
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  Local
    44.44.67.7 from 44.44.67.7 (172.16.77.77)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      rx pathid: 0, tx pathid: 0x0
```

- **valid** - means route is reachable, despite being remote.
- **internal** - means that's iBGP route.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

- **best** - means from all of routes available, this one was selected as the best one. Right now, it's obvious, as it's the only one available.

Then, we can check router global routing table for the entry from BGP protocol:

```
CEL6#sh ip route bgp
[…]

Gateway of last resort is not set

      40.0.0.0/24 is subnetted, 1 subnets
B        40.60.27.0 [200/0] via 44.44.67.7, 00:08:59
```

The [200/0] signifies default iBGP administrative distance (200) and metric (0) for the route. We can also check details of the BGP RIB-installed route:

```
CEL6#sh ip route 40.60.27.0
Routing entry for 40.60.27.0/24
  Known via "bgp 100", distance 200, metric 0, type internal
  Last update from 44.44.67.7 00:11:16 ago
  Routing Descriptor Blocks:
  * 44.44.67.7, from 44.44.67.7, 00:11:16 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
      MPLS label: none
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.1.4: Advertise network in BGP using ‘redistribute’ statement

There are multiple ways to advertise network into BGP. We already covered ‘**network**’ statement. Let’s look at redistributing routes into BGP.

Our current prefix is originated using ‘network’ statement, and has some default properties:

```
CEL6#sh ip bgp 40.60.27.0
BGP routing table entry for 40.60.27.0/24, version 5
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  Local
    44.44.67.7 from 44.44.67.7 (172.16.77.77)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      rx pathid: 0, tx pathid: 0x0
```

Let’s redistribute connected into BGP and check what will be the attributes:

```
CEL6(config)#router bgp 100
CEL6(config-router)#redistribute connected
CEL6(config-router)#^Z
```

Now we see more networks in the BGP:

```
CEL6#sh ip bgp
[…]
     Network          Next Hop            Metric LocPrf Weight Path
 *>   21.41.46.0/24    0.0.0.0                  0         32768 ?
 *>i  40.60.27.0/24    172.16.7.7               0    100      0 i
 *>   41.61.16.0/24    0.0.0.0                  0         32768 i
 *>   44.44.67.0/24    0.0.0.0                  0         32768 ?
 *>   172.16.6.6/32    0.0.0.0                  0         32768 ?
```

The routes have the same properties as loopback network introduced using ‘network’ command. For example, for 21.41.46.0/24:

```
CEL6#sh ip bgp 21.41.46.0
BGP routing table entry for 21.41.46.0/24, version 5
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     3
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (172.16.6.6)
      Origin incomplete, metric 0, localpref 100, weight 32768, valid, sourced, best
      rx pathid: 0, tx pathid: 0x0
```

We see that BGP keeps track of origin of the route (‘incomplete’ – as is normal for effect of ‘**redistribute**’ command), metric (0), and assigns some default Local Preference value (100). Additionally, we see that Cisco routers assign vendor-specific and router-specific value of ‘**weight**’ (32768). The route is also marked as ‘**sourced**’, which means it’s originated on this router.

While various Service Providers operate on different defaults, it’s accepted practice to modify the networks being injected into BGP as:

- originating from IGP – this helps in path selection stability
- having some default local preference – 100 is default for Cisco
- having some default metric – default is 0, but it’s per SP policy

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

We'll build route-map to help us redistribute routes in a controlled fashion in line with our requirements:

- originate from IGP
- assign Local Preference of 500

```
CEL6(config)#route-map RM-RED-BGP permit 10
CEL6(config-route-map)#set origin igp
CEL6(config-route-map)#set local-preference 500
```

Now, we will correct the redistribution statement under BGP process:

```
CEL6(config)#router bgp 100
CEL6(config-router)#redistribute connected route-map RM-RED-BGP
```

The BGP contents will be modified immediately:

```
CEL6#sh ip bgp
BGP table version is 10, local router ID is 172.16.6.6
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>   21.41.46.0/24    0.0.0.0                  0    500  32768 i
 *>i  40.60.27.0/24    172.16.7.7               0    100      0 i
 *>   41.61.16.0/24    0.0.0.0                  0         32768 i
 *>   44.44.67.0/24    0.0.0.0                  0    500  32768 i
 *>   172.16.6.6/32    0.0.0.0                  0    500  32768 i
```

And for the details of the example prefix of 21.41.46.0/24:

```
CEL6#sh ip bgp 21.41.46.0
BGP routing table entry for 21.41.46.0/24, version 8
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     3
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (172.16.6.6)
      Origin IGP, metric 0, localpref 500, weight 32768, valid, sourced, best
      rx pathid: 0, tx pathid: 0x0
```

Notice that because we're using both 'network' statement and 'route-map' redistribution method, network 41.61.16.0/24 is first hit by 'network' statement and not modified anymore with the 'redistribute' statement. We can fix this by removing 'network' advertisement and rely on redistribution:

```
CEL6(config)#router bgp 100
CEL6(config-router)#no network 41.61.16.0 mask 255.255.255.0
```

It's immediately reflected:

```
CEL6#sh ip bgp 41.61.16.0
BGP routing table entry for 41.61.16.0/24, version 12
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     3
  Refresh Epoch 1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Local
    0.0.0.0 from 0.0.0.0 (172.16.6.6)
      Origin IGP, metric 0, localpref 500, weight 32768, valid, sourced, best
      rx pathid: 0, tx pathid: 0x0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.1.5: Loopback peering in BGP

Now we will switch to Loopback interface peering. Please note, that for iBGP sessions hop count is not important and session will establish despite being multihop from BGP perspective:

```
CEL6(config)#router bgp 100
CEL6(config-router)#no neighbor 44.44.67.7
CEL6(config-router)#neighbor 172.16.7.7 remote-as 100
CEL6(config-router)#neighbor 172.16.7.7 update-source loopback 0
CEL6(config-router)#^Z

CEL7(config)#router bgp 100
CEL7(config-router)#no neighbor 44.44.67.6
CEL7(config-router)#neighbor 172.16.6.6 remote-as 100
CEL7(config-router)#neighbor 172.16.6.6 update-source loopback 0
CEL7(config-router)#^Z
```

The session should come up, this time using the loopback 0 address as sources for the sessions, and remote router loopback 0 address as destination. The connection is possible, as loopback interfaces are advertised by the OSPF protocol:

```
CEL6#sh ip route 172.16.7.7
Routing entry for 172.16.7.7/32
  Known via "ospf 1", distance 110, metric 2, type intra area
  Last update from 44.44.67.7 on GigabitEthernet3, 00:30:56 ago
  Routing Descriptor Blocks:
  * 44.44.67.7, from 172.16.77.77, 00:30:56 ago, via GigabitEthernet3
      Route metric is 2, traffic share count is 1

CEL7#sh ip route 172.16.6.6
Routing entry for 172.16.6.6/32
  Known via "ospf 1", distance 110, metric 4, type intra area
  Last update from 40.60.27.2 on GigabitEthernet2, 00:28:55 ago
  Routing Descriptor Blocks:
  * 40.60.27.2, from 172.16.6.6, 00:28:55 ago, via GigabitEthernet2
      Route metric is 4, traffic share count is 1
```

However, the remote prefixes are reachable via recursive resolution with CEF:

```
CEL6#sh ip cef 40.60.27.0 detail
40.60.27.0/24, epoch 2, flags [rib only nolabel, rib defined all labels]
  recursive via 172.16.7.7
    nexthop 44.44.67.7 GigabitEthernet3
```



From now on, we will use **Loopbacks to configure BGP sessions unless otherwise specified**. Please note that while providing stability it requires underlying IGP visibility between different ASes. Also, this may mean in rare eBGP situations, that the session will establish but around failed/not advertised link, which may not be what You want.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The quickest way in modern IOS versions and IOS-XE to check for the whole router BGP config is to do running-config parsing via ‘section’ module, like this:

```
CEL6#sh running-config | sect router bgp
router bgp 100
 bgp log-neighbor-changes
 network 41.61.16.0 mask 255.255.255.0
 neighbor 172.16.7.7 remote-as 100
 neighbor 172.16.7.7 update-source Loopback0
CEL7#sh running-config | sect router bgp
router bgp 100
 bgp log-neighbor-changes
 network 40.60.27.0 mask 255.255.255.0
 neighbor 172.16.6.6 remote-as 100
 neighbor 172.16.6.6 update-source Loopback0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: ASPLAIN and ASDOT notations in IOS/IOS-XE

Task 1.2.1: eBGP session over loopbacks

Reconfigure R7 to be in BGP routing AS 10.10. Also, please remember that eBGP sessions require eBGP multihop configuration if the peers are not on the same link:

```
CEL7#conf t
CEL7(config)#no router bgp 100
CEL7(config)#router bgp 10.10
CEL7(config-router)#network 40.60.27.0 mask 255.255.255.0
CEL7(config-router)#neighbor 172.16.6.6 remote-as 100
CEL7(config-router)#neighbor 172.16.6.6 update-source Loopback0
CEL7(config-router)#neighbor 172.16.6.6 ebgp-multihop 5
CEL7(config-router)#^Z
CEL7#

CEL6#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
CEL6(config)#router bgp 100
CEL6(config-router)#neighbor 172.16.7.7 remote-as 10.10
CEL6(config-router)#neighbor 172.16.7.7 update-source Loopback0
CEL6(config-router)#neighbor 172.16.7.7 ebgp-multihop 5
CEL6(config-router)#^Z
CEL6#
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2.2: Changing between ASPLAIN and ASDOT

Notice, that by default IOS and IOS-XE use BGP **asplain** notation. This makes the whole 4-byte ASN appear as big integer:

```
CEL6#sh ip bgp summary
BGP router identifier 172.16.6.6, local AS number 100
[…]

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.7.7      4       655370       6       6        4    0    0 00:00:54        1

CEL7#sh ip bgp summary
BGP router identifier 172.16.77.77, local AS number 655370
```

This facilitates clear and strict enforcement of AS_PATH filtering regular expressions, as all ASNs are simply integer numbers.

You can switch representation of the ASNs to **asdot**, bear in mind however, that this **will influence the way regular expressions** are processed on the AS_PATH. To change the representation issue following command:

```
CEL7(config)#router bgp 10.10
CEL7(config-router)#bgp asnotation dot
```

The output will immediately change:

```
CEL7#sh ip bgp summary
BGP router identifier 172.16.77.77, local AS number 10.10
```

Let's do the same on CEL6:

```
CEL6(config)#router bgp 100
CEL6(config-router)#bgp asnotation dot
```

Now, CEL6 will properly reflect the **asdot** format for any 4B ASN:

```
CEL6#sh ip bgp summary
BGP router identifier 172.16.6.6, local AS number 100
[…]

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.7.7      4        10.10       6       6        4    0    0 00:00:54        1
```



If you need to do number transition between ASPLAIN to ASDOT by hand, you can use this simple process.

Assuming we want to change ASPLAIN ASN of 394189 to ASDOT, here's the example:

1. ASPLAIN = **394189**
2. 394189 / 65535 = 6 (we're taking only integer part)
3. 394189 – (65535*6) = 979
4. 979 – 6 = 973
5. ASDOT = **6.973**

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3: Filtering advertisements in IOS and IOS-XE

You can use following tools to filter the advertised and received routes along with their attributes:

For **inbound** updates in order of execution:

- route-map (most flexible way of processing the updates – recommended)
- filter-list (AS_PATH access-list)
- prefix-list
- distribute-list (Extended ACLs, matching on the prefixes received)

For **outbound** updates in order of execution:

- filter-list
- route-map | unsuppress-map
- advertise-map (conditional-advertisement)
- prefix-list
- distribute-list
- ORF prefix-list (prefix list sent from neighbor)

While the CCIE SP should master all ways of configuring the filtering, best practice is to use ONLY route-maps for both inbound and outbound prefix manipulation, unless some specific feature is needed (like ORF for example).

Task 1.3.1: Using route-maps to filter updates in BGP

Let's build a sample route-map to filter incoming updates and check for regular expression of 10.10 (which will become "10\.10" to escape the special dot meaning):

```
CEL6(config)#ip as-path access-list 10 permit ^10\.10$
CEL6(config)#route-map RM-BGP-IN permit 10
CEL6(config-route-map)#match as-path 10
```



route-maps are processed sequentially. First hit (match) stops processing unless '**continue**' keyword is added.

The '**permit**' in route-map definition before the index means prefixes matching specific requirements in the '**match**' section will be allowed to be processed by routing protocol. '**deny**' means prefixes matching the requirements won't be processed. Some caveats apply however here with some specific uses of route-maps – which will be covered later.

At the end of route-map there's always 'deny' entry (default), which once route-map is applied to routing function, will skip processing other updates. If you don't permit any other prefixes, they won't be processed.

We will now use the route-map as input filter for the incoming updates from on CEL6 from CEL7 (our BGP in ASN 10.10 in asdot notation):

```
CEL6(config)#router bgp 100
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CEL6(config-router)#neighbor 172.16.7.7 route-map RM-BGP-IN in
```

After refreshing the session, we can see that the CEL6 is still using the prefix as it went through the policy and was accepted:

```
CEL6#sh ip bgp neighbors 172.16.7.7 routes
[…]

     Network          Next Hop            Metric LocPrf Weight Path
 *>   40.60.27.0/24    172.16.7.7               0             0 10.10 i

Total number of prefixes 1
```

Now, let's change the ASN notation from **asdot** to **asplain** on CEL6, refresh the session again, and turn on debug to check the route-map in action:

```
CEL6#clear ip bgp *
*Jan 1 23:38:51.588: %BGP-5-ADJCHANGE: neighbor 172.16.7.7 Up
*Jan 1 23:38:51.592: BGP(0): 172.16.7.7 rcvd UPDATE w/ attr: nexthop 172.16.7.7, origin
i, metric 0, merged path 655370, AS_PATH
*Jan 1 23:38:51.592: BGP(0): 172.16.7.7 rcvd 40.60.27.0/24 -- DENIED due to: route-map;
```

Despite CEL7 advertising the route:

```
CEL7#sh ip bgp neighbors 172.16.6.6 advertised-routes
[…]

     Network          Next Hop            Metric LocPrf Weight Path
 *>   40.60.27.0/24    0.0.0.0                  0         32768 i
```

CEL6 is not accepting the route:

```
CEL6#sh ip bgp summary
[…]
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.7.7      4       655370       7       7        2    0    0 00:02:23        0
```

We can see that the route was denied also in the summary of the BGP neighbor information:

```
CEL6#sh ip bgp neighbors 172.16.7.7
[…]
 For address family: IPv4 Unicast
[…]
                                   Outbound    Inbound
  Local Policy Denied Prefixes:    --------    -------
    route-map:                            0          1
    Total:                                0          1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3.2: Using filter-list to filter updates in BGP

The same effect could be achieved using the **filter-list** filtering method:

```
CEL6(config-router-af)#neighbor 172.16.7.7 filter-list 10 in
```

However, this way is less flexible with capability only to pass or reject routes only based on the AS_PATH attribute.

Task 1.3.3: Reconfigure AS100 back to original configuration

Reconfigure CEL7 back again to AS100 and for loopback0 based iBGP peering between those two routers.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4: BGP Address Families in IOS and IOS-XE

By default, IOS and IOS-XE parser assumes “old” style of configuring the networks under BGP. Default is to advertise IPv4 AF and nothing else.

To change, or “upgrade” the BGP CLI we can either just start to configure other AFs, like IPv6 unicast (for example), or request upgrading the parser:

```
CEL6(config-router)#bgp upgrade-cli
You are about to upgrade to the hierarchical AFI syntax of bgp commands

Are you sure ? [yes]: yes
CEL6(config-router)#
```

From now moment on per-AF configuration, like advertising of networks and policies will be configured under separate address-family sections, like that:

```
CEL6(config)#router bgp 100
CEL6(config-router)#neighbor 172.16.7.7 remote-as 100
CEL6(config-router)#neighbor 172.16.7.7 update-source loopback 0
CEL6(config-router)#address-family ipv4 unicast
CEL6(config-router-af)#neighbor 172.16.7.7 activate
CEL6(config-router-af)#network 41.61.16.0 mask 255.255.255.0
CEL6(config-router-af)#^Z
```

For final config on CEL6:

```
CEL6#sh running-config | sect router bgp
router bgp 100
 bgp log-neighbor-changes
 neighbor 172.16.7.7 remote-as 100
 neighbor 172.16.7.7 update-source Loopback0
 !
 address-family ipv4
  network 41.61.16.0 mask 255.255.255.0
  neighbor 172.16.7.7 activate
 exit-address-family
```

And the same for CEL7:

```
CEL7(config)#router bgp 100
CEL7(config-router)#bgp upgrade-cli
You are about to upgrade to the hierarchical AFI syntax of bgp commands
Are you sure ? [yes]: yes
CEL7(config-router)#neighbor 172.16.6.6 remote-as 100
CEL7(config-router)#neighbor 172.16.6.6 update-source loopback 0
CEL7(config-router)#address-family ipv4 unicast
CEL7(config-router-af)#network 40.60.27.0 mask 255.255.255.0
CEL7(config-router-af)#neighbor 172.16.6.6 activate
CEL7(config-router-af)#^Z
```

For the config of:

```
CEL7#sh running-config | sect router bgp
router bgp 100
 bgp log-neighbor-changes
 neighbor 172.16.6.6 remote-as 100
 neighbor 172.16.6.6 update-source Loopback0
 !
 address-family ipv4
  network 40.60.27.0 mask 255.255.255.0
  neighbor 172.16.6.6 activate
 exit-address-family
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4.1: IPv6 unicast session configuration in BGP

Now, we'll add IPv6 session and exchange the same links as for IPv4:

```
CEL7(config)#router bgp 100
CEL7(config-router)#neighbor 2001:44:66:CC::6 remote-as 100
CEL7(config-router)#neighbor 2001:44:66:CC::6 update-source loopback 0
CEL7(config-router)#address-family ipv6 unicast
CEL7(config-router-af)#network 2001:40:60:17::/64
CEL7(config-router-af)#neighbor 2001:44:66:CC::6 activate
CEL7(config-router-af)#^Z
```

And for CEL6:

```
CEL6(config)#router bgp 100
CEL6(config-router)#neighbor 2001:44:66:CC::7 remote-as 100
CEL6(config-router)#neighbor 2001:44:66:CC::7 update-source loopback 0
CEL6(config-router)#address-family ipv6 unicast
CEL6(config-router-af)#neighbor 2001:44:66:CC::7 activate
CEL6(config-router-af)#network 2001:41:61:16::/64
CEL6(config-router-af)#^Z
```

Separate, IPv6 transport session was established and the prefixes were exchanged:

```
CEL6#sh ip bgp ipv6 unicast summary
[…]
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001:44:66:CC::7
                4          100       7       6        5    0    0 00:01:16        1
```

The prefix is installed in the IPv6 RIB:

```
CEL6#sh ipv6 route bgp
IPv6 Routing Table - default - 16 entries
[…]
B   2001:40:60:17::/64 [200/0]
     via 2001:44:66:CC::7
```

The same is happening on CEL7:

```
CEL7#sh ip bgp ipv6 unicast
[…]
     Network          Next Hop            Metric LocPrf Weight Path
 *>   2001:40:60:17::/64
                      ::                       0         32768 i
 *>i  2001:41:61:16::/64
                      2001:44:66:CC::6
                                               0    100      0 i
```

Notice, that by default, IOS and IOS-XE will keep the transport sessions separate for their own AFs:

```
CEL6#sh running-config | sect router bgp
router bgp 100
 bgp log-neighbor-changes
 neighbor 2001:44:66:CC::7 remote-as 100
 neighbor 2001:44:66:CC::7 update-source Loopback0
 neighbor 172.16.7.7 remote-as 100
 neighbor 172.16.7.7 update-source Loopback0
 !
 address-family ipv4
  network 41.61.16.0 mask 255.255.255.0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  no neighbor 2001:44:66:CC::7 activate
  neighbor 172.16.7.7 activate
 exit-address-family
 !
 address-family ipv6
  network 2001:41:61:16::/64
  neighbor 2001:44:66:CC::7 activate
 exit-address-family
```

From the IPv6 BGP neighbor:

```
CEL6#sh ip bgp ipv6 unicast neighbors 2001:44:66:CC::7
BGP neighbor is 2001:44:66:CC::7,  remote AS 100, internal link
  BGP version 4, remote router ID 172.16.77.77
  BGP state = Established, up for 00:05:39
[…]
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv6 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
```



Please take a note, that best practice is to keep the address families separated to their own transport sessions.

Trying to advertise IPv6 over IPv4 (for example) has it's caveats. We will cover such configuration options in Session 4.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.5: Configure routers in AS 3356

Configure IOS XR routers R1, R2, R3 and IOS-XE router R5 to peer using loopback interfaces in full mesh.

For R1 configuration we will use the normal way of configuring the BGP sessions. For the next routers we will use IOS XR “BGP Templates” feature which speeds up boring way of configuring the BGP neighbors and session options at the same time.

Task 1.5.1: Simple iBGP session between R1 and R2, R3 and R5

We will advertise the Loopback0 address over the BGP from R1 towards it’s BGP peers in AS3356 by means of ‘**network**’ statement. Notice the IPv4 (and any other AF) statements are configured globally, and then only specific AFs are assigned to neighbors in IOS XR – which is what new version of IOS/IOS-XE parser tries to imitate:

```
RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-af)#network 33.56.1.1/32
RP/0/0/CPU0:SP1R1(config-bgp)#neighbor 33.56.2.2
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#remote-as 3356
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#neighbor 33.56.3.3
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#remote-as 3356
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#neighbor 33.56.5.5
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#remote-as 3356
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes

RP/0/0/CPU0:SP1R1#sh run router bgp
router bgp 3356
 address-family ipv4 unicast
  network 33.56.1.1/32
 !
 neighbor 33.56.2.2
  remote-as 3356
  update-source Loopback0
  address-family ipv4 unicast
  !
 !
 neighbor 33.56.3.3
  remote-as 3356
  update-source Loopback0
  address-family ipv4 unicast
  !
 !
 neighbor 33.56.5.5
  remote-as 3356
  update-source Loopback0
  address-family ipv4 unicast
  !
 !
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

For testing, let's configure SP1R2 and check if the session goes up:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-af)#network 33.56.2.2/32
RP/0/0/CPU0:SP1R2(config-bgp-af)#exit
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 33.56.1.1
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#remote-as 3356
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
RP/0/0/CPU0:SP1R2#
```

If the routers are correctly connected, the session should come up almost immediately:

```
RP/0/0/CPU0:SP1R1#sh ip bgp sum
BGP router identifier 33.56.1.1, local AS number 3356
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 4
BGP main routing table version 4
BGP NSR Initial initsync version 2 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.


Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker               4          4          4          4           4           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
33.56.2.2         0  3356       4       4        4    0    0 00:00:23          1
33.56.3.3         0  3356       0       0        0    0    0 00:00:00 Active
33.56.5.5         0  3356       0       0        0    0    0 00:00:00 Active
```

…and from SP1R2 perspective:

```
RP/0/0/CPU0:SP1R2#sh ip bgp sum
BGP router identifier 33.56.2.2, local AS number 3356
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 4
BGP main routing table version 4
BGP NSR Initial initsync version 2 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.

Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker               4          4          4          4           4           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
33.56.1.1         0  3356       5       5        4    0    0 00:01:33          1

RP/0/0/CPU0:SP1R2#sh ip bgp
BGP router identifier 33.99.99.99, local AS number 3356
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 4
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
BGP main routing table version 4
BGP NSR Initial initsync version 2 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*>i33.56.1.1/32       33.56.1.1                0    100      0 i
*> 33.56.2.2/32       0.0.0.0                  0         32768 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.5.2: BGP router-id selection



By default, IOS XR select the highest-numbered loopback interface as BGP router-id if there is saved configuration, or primary IPv4 address configured on first loopback interface in configuration mode. Note, that IOS XR won't select IPv4 address for BGP router-id from physical interfaces!

It is recommended to configure Loopback within own IP address space per each router solely for the purpose of assigning stable router-id's. Properly executed it has a benefit of being able to identifying the routers at the same time.

Router SP1R1 has one IP on Loopback0: 33.56.1.1.

Let's assume SP1R2 would have Loopback1 interface that has IP higher than of Loopback0 and destroys our clean addressing scheme for router-id assignment:

```
RP/0/0/CPU0:SP1R2#sh ip int brief
Interface                      IP-Address      Status          Protocol Vrf-Name
Loopback0                      33.56.2.2       Up              Up       default
Loopback1                      33.99.99.99     Up              Up       default
MgmtEth0/0/CPU0/0              10.255.0.249    Up              Up       Mgmt-intf
GigabitEthernet0/0/0/0         33.56.23.2      Up              Up       default
GigabitEthernet0/0/0/1         40.60.27.2      Up              Up       default
GigabitEthernet0/0/0/2         33.56.12.2      Up              Up       default
GigabitEthernet0/0/0/3         41.61.26.2      Shutdown        Down     default
GigabitEthernet0/0/0/4         33.56.24.2      Shutdown        Down     default
```

This makes BGP pick up 33.99.99.99 as router-id which may not be optimal:

```
RP/0/0/CPU0:SP1R2#sh ip bgp | i router
BGP router identifier 33.99.99.99, local AS number 3356
```

We'll force the change of router-id for SP1R2 – this will cause all sessions to reset:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#bgp router-id 33.56.2.2
RP/0/0/CPU0:SP1R2(config-bgp)#

RP/0/0/CPU0:SP1R2#sh ip bgp sum | i 33.56.1.1
33.56.1.1          0  3356      12      11         0    0    0 00:00:03        1
```

Now the SP1R2 has 'proper' router-id from our address plan:

```
RP/0/0/CPU0:SP1R2#sh ip bgp sum | i router
BGP router identifier 33.56.2.2, local AS number 3356
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.5.3: BGP neighbor status and session options in IOS XR

As was with the case with IOS and IOS-XE, IOS XR shows multiple neighbor-related data when asked for specific node:

```
RP/0/0/CPU0:SP1R1#sh bgp ipv4 unicast neighbor 33.56.2.2

BGP neighbor is 33.56.2.2
 Remote AS 3356, local AS 3356, internal link
 Remote router ID 33.56.2.2
  BGP state = Established, up for 00:03:15
  NSR State: None
  Last read 00:00:10, Last read before reset 00:04:15
  Hold time is 180, keepalive interval is 60 seconds
  Configured hold time: 180, keepalive: 60, min acceptable hold time: 3
[…]
  Multi-protocol capability received
  Neighbor capabilities:
    Route refresh: advertised (old + new) and received (old + new)
    4-byte AS: advertised and received
    Address family IPv4 Unicast: advertised and received
[…]
 For Address Family: IPv4 Unicast
[…]
  1 accepted prefixes, 1 are bestpaths
  Cumulative no. of prefixes denied: 0.
  Prefix advertised 1, suppressed 0, withdrawn 0
  Maximum prefixes allowed 1048576
  Threshold for warning message 75%, restart interval 0 min
  AIGP is enabled
[…]
  Local host: 33.56.1.1, Local port: 60397, IF Handle: 0x00000000
  Foreign host: 33.56.2.2, Foreign port: 179
  Last reset 00:03:43, due to Peer closing down the session
  Peer reset reason: Remote closed the session (Connection timed out)
  Time since last notification sent to neighbor: 00:10:19
  Error Code: hold time expired
  Notification data sent:
    None
```

Notice, that by default IOS XR won't show up TCP MSS setting. In troubleshooting problems this may be one of things to check. The best way to check the TCP MSS status is to go through the TCP Protocol Control Block that's allocated for this specific neighbor. First, we need to check the PCB that's used for this specific neighbor:

```
RP/0/0/CPU0:SP1R1#sh tcp brief | utility egrep "PCB|33.56.2.2"
PCB      VRF-ID     Recv-Q Send-Q Local Address          Foreign Address        State
0x12159bc8 0x60000000      0      0  33.56.1.1:60397        33.56.2.2:179          ESTAB
```

Then we're checking the details of 0x12159bc8:

```
RP/0/0/CPU0:SP1R1#sh tcp detail pcb 0x12159bc8 | i MSS
Datagrams (in bytes): MSS 1240, peer MSS 1240, min MSS 1240, max MSS 1240
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Notice, that IOS XR uses by default 1240 bytes instead of maximal value of 1460. Once we turn on TCP Path MTU Discovery on both nodes (SP1R1 and SP1R2), we'll get the 1460:

```
RP/0/0/CPU0:SP1R1(config)#tcp path-mtu-discovery
RP/0/0/CPU0:SP1R1(config)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes

RP/0/0/CPU0:SP1R2(config)#tcp path-mtu-discovery
RP/0/0/CPU0:SP1R2(config)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
RP/0/0/CPU0:SP1R2#clear ip bgp *
```

Now, let's check SP1R1 reported values for TCP MSS:

```
RP/0/0/CPU0:SP1R1#sh tcp detail pcb 0x1215a54c | i MSS
Datagrams (in bytes): MSS 1460, peer MSS 1460, min MSS 1460, max MSS 1460
```



While the difference between 1460 and the default for IOS XR of 1240 is "only" 220 bytes, please make note that in typical Service Provider networks Jumbo Frames are used, up to the 9kB. BGP still will use only 4096 octets (bytes) (maximum BGP message size as per RFC 4271), but the benefits of working PMTUD will be clearly visible in convergence times.

IOS XR also has useful helper to check for number and actual advertised prefixes, while IOS and IOS-XE can only show advertised routes.

For IOS XR, to check for specific count of prefixes advertised to neighbor do:

```
RP/0/0/CPU0:SP1R1#sh bgp ipv4 unicast neighbors 33.56.2.2 advertised-count
No of prefixes Advertised: 1
```

…and to check for specific prefixes:

```
RP/0/0/CPU0:SP1R1#sh bgp ipv4 unicast neighbors 33.56.2.2 advertised-routes
Network            Next Hop        From            AS Path
33.56.1.1/32       33.56.1.1       Local           i
Processed 1 prefixes, 1 paths
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.5.4: BGP route entry in IOS XR

The BGP entry for this remote prefix can be viewed on IOS XR by executing:

```
RP/0/0/CPU0:SP1R1#sh ip bgp 33.56.2.2
BGP routing table entry for 33.56.2.2/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                14          14
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    33.56.2.2 (metric 2) from 33.56.2.2 (33.56.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal, best, group-best
      Received Path ID 0, Local Path ID 0, version 14
```

Apart from IOS XR specific speaker process information section, we can see basically the same data as in IOS and IOS-XE – origin and path information:

- **valid** - means route is reachable, despite being remote.
- **internal** - means that's iBGP route.
- **best** - means from all of routes available, this one was selected as the best one. Right now it's obvious, as it's the only one available.

As the loopbacks are installed already via OSPF, BGP with higher administrative distance won't be able to install them in router RIB:

```
RP/0/0/CPU0:SP1R1#sh ip route bgp

% No matching routes found
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.5.5: Advertise network in BGP using ‘redistribute’ statement

As is the case with IOS and IOS-XE, IOS XR has also multiple ways to advertise network into BGP. We already covered ‘**network**’ statement. Let’s look at redistributing routes into BGP.

Let’s redistribute connected into BGP and check what will be the attributes:

```
RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-af)#redistribute connected
RP/0/0/CPU0:SP1R1(config-bgp-af)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
RP/0/0/CPU0:SP1R1#
```

Now we see more networks in the BGP:

```
RP/0/0/CPU0:SP1R1#sh ip bgp
[…]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 33.56.1.1/32       0.0.0.0                  0         32768 i
*>i33.56.2.2/32       33.56.2.2                0    100      0 i
*> 33.56.12.0/24      0.0.0.0                  0         32768 ?
*> 33.56.14.0/24      0.0.0.0                  0         32768 ?
*> 33.56.15.0/24      0.0.0.0                  0         32768 ?
*> 41.61.16.0/24      0.0.0.0                  0         32768 ?

Processed 6 prefixes, 6 paths
```

The routes have the same properties as loopback network introduced using ‘network’ command. For example, for 33.56.12.0/24:

```
RP/0/0/CPU0:SP1R1#sh ip bgp 33.56.12.0
BGP routing table entry for 33.56.12.0/24
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 17          17
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    33.56.2.2
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    33.56.2.2
  Local
    0.0.0.0 from 0.0.0.0 (33.56.1.1)
      Origin incomplete, metric 0, localpref 100, weight 32768, valid, redistributed,
best, group-best
      Received Path ID 0, Local Path ID 0, version 17
```

On our neighbor the prefixes are visible as well:

```
RP/0/0/CPU0:SP1R2#sh ip route bgp

B    33.56.14.0/24 [200/0] via 33.56.1.1, 00:01:56
B    33.56.15.0/24 [200/0] via 33.56.1.1, 00:01:56
B    41.61.16.0/24 [200/0] via 33.56.1.1, 00:01:56
```

IOS XR uses powerful RPL language to process routing information transformations. We’ll build simple route-policy to reflect the changes we did

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

with IOS/IOS-XE to the BGP redistribute information – including changing origin to IGP and local preference to 500.

The construction for this simplistic route-policy is pretty similar to that of route-maps:

```
RP/0/0/CPU0:SP1R1(config)#route-policy BGP-RED-MODIFY
RP/0/0/CPU0:SP1R1(config-rpl)#set origin igp
RP/0/0/CPU0:SP1R1(config-rpl)#set local-preference 500
RP/0/0/CPU0:SP1R1(config-rpl)#end-policy
RP/0/0/CPU0:SP1R1(config)#
```

Once route-policy is committed, we can check if it's in the running configuration:

```
RP/0/0/CPU0:SP1R1#sh running-config route-policy BGP-RED-MODIFY
route-policy BGP-RED-MODIFY
  set origin igp
  set local-preference 500
end-policy
RP/0/0/CPU0:SP1R1#sh rpl route-policy states

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

The following policies are (ACTIVE)
-----------------------------------------
route-policy From-AS100
[…]

The following policies are (INACTIVE)
-----------------------------------------
None found with this status.

The following policies are (UNUSED)
-----------------------------------------
route-policy BGP-PREPEND
[…]
```



You can debug RPL policy with the '**debug bgp policy-execution events**' command.

Let's apply it to the redistribution command:

```
RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-af)#redistribute connected route-policy ?
  BGP-RED-MODIFY  Name of the policy ! notice parser is proposing existing policies
  WORD            Name of the policy
RP/0/0/CPU0:SP1R1(config-bgp-af)#redistribute connected route-policy BGP-RED-MODIFY
RP/0/0/CPU0:SP1R1(config-bgp-af)#
```

Again, we see that the network originated with the '**network**' command has it's attributes unaffected, but all other networks, redistributed from connected, have been modified:

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R1#sh ip bgp
[…]
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 33.56.1.1/32       0.0.0.0                  0         32768 i
*>i33.56.2.2/32       33.56.2.2                0    100      0 i
*> 33.56.12.0/24      0.0.0.0                  0    500  32768 i
*> 33.56.14.0/24      0.0.0.0                  0    500  32768 i
*> 33.56.15.0/24      0.0.0.0                  0    500  32768 i
*> 41.61.16.0/24      0.0.0.0                  0    500  32768 i
```

Let's check the 33.56.12.0/24 prefix again:

```
RP/0/0/CPU0:SP1R1#sh ip bgp 33.56.12.0
BGP routing table entry for 33.56.12.0/24
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 21          21
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    33.56.2.2
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    33.56.2.2
  Local
    0.0.0.0 from 0.0.0.0 (33.56.1.1)
      Origin IGP, metric 0, localpref 500, weight 32768, valid, redistributed, best,
group-best
      Received Path ID 0, Local Path ID 0, version 21
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.6: Authenticating BGP sessions

We will now configure BGP authentication options.

IGP protocols and BGP can use MD5 as authentication mechanism, but it's already obsoleted, and shouldn't be typically used.



Practically speaking, if attacker is able to access your internal, core network, and sniff traffic or inject traffic into that, MD5 will be least of your problems.

BUT, sane security and layered defense means that even potentially "non-realistic" scenarios need to be addressed. "Non-realistic" scenarios are first attack vector 9 times out of 10.

New mechanism that enables more flexible approach to authentication hashes is TCP-AO. In IOS XR, it's done using keychains.

Task 1.6.1: Using BGP with MD5 with IOS-XE and IOS XR

Configuration of MD5 based BGP authentication is straightforward. You just add '**password**' stance to neighbor definition and paste either clear-text password (by default, will be auto-encrypted using well-known and reversible Cisco "7" password algorithm), or already "hashed" form of password, like this:

```
SP1R5-RR#conf t
SP1R5-RR(config)#router bgp 3356
SP1R5-RR(config-router)#neighbor 33.56.1.1 password AS3356
```

Note that this will stay in unencrypted form, unless '**service password-encryption**' is enabled on global level in IOS/IOS-XE:

```
SP1R5-RR#conf t
SP1R5-RR(config)#service password-encryption
SP1R5-RR(config)#^Z
SP1R5-RR#sh running-config |  s router bgp
router bgp 3356
[…]
 neighbor 33.56.1.1 remote-as 3356
 neighbor 33.56.1.1 password 7 00252055570E5D
```

On SP1R1 you obviously need to match the password, so the session can proceed without failing the hash check:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#neighbor 33.56.5.5
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#password clear AS3356
```

You can verify the MD5 is used as hashing by displaying details of BGP session on IOS-XE side of connection:

SP1R5-RR#sh bgp ipv4 unicast neighbors 33.56.1.1 | i md5

Option Flags: nagle, path mtu capable, md5, Retrans timeout

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

On IOS XR, depending on version used, the procedure is more involved.

In IOS XR versions older than 6.5.x, the only visible sign that given TCP session uses authentication, will be to query directly TCP PCB (Protocol Control Block). Given we know we're looking for 33.56.5.5 as neighbor address, first step to display PCB is to check existing TCP connections:

```
RP/0/0/CPU0:SP1R1#sh tcp brief | utility egrep "PCB|33.56.5.5"
   PCB      VRF-ID     Recv-Q Send-Q Local Address          Foreign Address        State
0x1217fc7c 0x60000000      0      0  33.56.1.1:646          33.56.5.5:30201        ESTAB
0x1217477c 0x60000000      0      0  33.56.1.1:49657        33.56.5.5:179          ESTAB
```

Now, we know the corresponding PCB value for session terminated at port 179/tcp (BGP), we can query PCB directly by identifier:

```
RP/0/0/CPU0:SP1R1#sh tcp pcb 0x1217477c | i MD5
Feature flags: MD5, Nagle
```

In IOS XR versions after 6.5.x, there's extended view for MD5 authentication available in PCB. Here's sample output:

```
RP/0/0/CPU0:SP1R1#sh tcp detail pcb 0x00007fef84008358 | utility egrep "auth|MD5|Peer"
Feature flags: MD5, Win Scale, Nagle
Num of peers with authentication info: 1
OBJ-ID      Peer                    WKP  VRF-ID      Type  Auth-Info-Checksum
0x40002fa0  10.0.0.22/32             BGP  0x60000000  MD5   0x00006b09
```

Delete MD5 authentication before we move to the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.6.2: Using BGP with TCP-AO in IOS XR

For use of TCP-AO, we need first to define keychain containing both keys (passwords) and type of hash we want to use.



Note that versions used in the lab, TCP-AO is not yet supported. TCP-AO support for IOS-XE based routers was introduced in 16.12(1) version.

For virtual routers in lab, only MD5 is typically available, because they don't run cryptographic license. For production network, SHA-1 or SHA-256 seems to be good candidate for now.



Note that for lab XRv routers, only HMAC-MD5 is supported as of today. To use any other crypto hash algorithms, please use XRv9000 fully crypto licensed routers.

While you can configure different algorithms, following error will be printed on console if you do so:

```
%IP-TCP-3-EAOPT_SEND_UNSUPPORTED_MAC : Unsupported MAC
algorithm(4) for keychain 'bgp-keys' (send-key 1) for peer
X.X.X.X
```

Here's example of configuration of keychain for BGP on IOS XR device:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#key chain bgp-keys
RP/0/0/CPU0:SP1R1(config-bgp-keys)#key 1
RP/0/0/CPU0:SP1R1(config-bgp-keys-1)#accept-lifetime 00:00:00 january 01 2017 $
RP/0/0/CPU0:SP1R1(config-bgp-keys-1)#key-string cisco.123
RP/0/0/CPU0:SP1R1(config-bgp-keys-1)#send-lifetime 00:00:00 january 01 2017 in$
RP/0/0/CPU0:SP1R1(config-bgp-keys-1)#cryptographic-algorithm ?
  HMAC-MD5       Configure HMAC-MD5 as cryptographic algorithm
  HMAC-SHA1-12   Configure HMAC-SHA1-12 as cryptographic algorithm
  HMAC-SHA1-20   Configure HMAC-SHA1-20 as cryptographic algorithm
  MD5            Configure MD5 as cryptographic algorithm ! only one supported for lab!
  SHA-1          Configure SHA-1-20 as cryptographic algorithm
RP/0/0/CPU0:SP1R1(config-bgp-keys-1)#cryptographic-algorithm hmac-md5
RP/0/0/CPU0:SP1R1(config-bgp-keys-1)#exit
RP/0/0/CPU0:SP1R1(config-bgp-keys)#accept-tolerance infinite
```

And then configure BGP session with keychain just created (we'll use direct session between SP1R1 and SP1R2 – note, Loopback0 should be visible in IGP already by that point, if not – please use physical interfaces):

```
RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#neighbor 33.56.2.2
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#remote-as 3356
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#keychain bgp-keys
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

If configuration is made correctly, after clearing up BGP sessions, they'll come up using keychain configuration. Same verification procedure needs to be used if you want to check if MD5 is used as authentication.

For IOS XR versions newer than 6.5.x, the authentication field for TCP PCB detail view will show EA however in place of MD5 to signify "Enhanced Authentication" being used:

```
RP/0/RP0/CPU0:SP1R1#sh tcp detail pcb 0x00007fef840079f8

================================================================
Connection state is ESTAB, I/O status: 0, socket status: 0
Established at Wed May 19 19:11:07 2017

PCB 0x7fef840079f8, SO 0x7fef84007548, TCPCB 0x7fef84007c78, vrfid 0x60000000,
Pak Prio: Medium, TOS: 192, TTL: 255, Hash index: 364
Local host: 10.0.0.18, Local port: 179 (Local App PID: 1055)
Foreign host: 10.0.0.22, Foreign port: 52369
(Local App PID/instance/SPL_APP_ID: 1055/1/0)

[…]

State flags: none
Feature flags: MD5, Win Scale, Nagle
Request flags: Win Scale

[…]

Num of peers with authentication info: 1
OBJ-ID      Peer                     WKP  VRF-ID      Type  Auth-Info-Checksum
0x40002fa0  10.0.0.22/32              BGP  0x60000000  EA    0x00006f68
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.7: ASPLAIN and ASDOT notations in IOS/IOS XR

Task 1.7.1: Changing between ASPLAIN and ASDOT

We will use commit and rollback feature of IOS XR to save our current configuration before moving forward. This will save us time in next sections.

After deleting the 'redistribute' command from under IPv4 AF of BGP 3356 at SP1R1, please commit the config labelling it with "AS3356":

```
RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp-af)#no redistribute connected route-policy BGP-RED-MODIFY
RP/0/0/CPU0:SP1R1(config-bgp-af)#commit label AS3356
RP/0/0/CPU0:SP1R1(config)#
```

As was the case with IOS and IOS-XE, IOS XR by default uses BGP **asplain** notation.

We will reconfigure SP1R1 to belong to AS 10.10:

```
RP/0/0/CPU0:SP1R1(config)#no router bgp 3356
RP/0/0/CPU0:SP1R1(config)#commit
RP/0/0/CPU0:SP1R1(config)#router bgp 10.10
RP/0/0/CPU0:SP1R1(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-af)#network 33.56.1.1/32
RP/0/0/CPU0:SP1R1(config-bgp-af)#exit
RP/0/0/CPU0:SP1R1(config-bgp)#neighbor 33.56.2.2
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#remote-as 3356
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#exit
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#ebgp-multihop 5
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#
```

Which brings us to SP1R1 being configured in this way:

```
RP/0/0/CPU0:SP1R1#sh running-config router bgp
router bgp 10.10
 address-family ipv4 unicast
  network 33.56.1.1/32
 !
 neighbor 33.56.2.2
  remote-as 100
  ebgp-multihop 5
  update-source Loopback0
  address-family ipv4 unicast
  !
 !
!
```

Now notice, how BGP ASN is displayed:

```
RP/0/0/CPU0:SP1R1#sh ip bgp sum | i router
BGP router identifier 33.56.1.1, local AS number 655370
```

This facilitates clear and strict enforcement of AS_PATH filtering regular expressions, as all ASNs are simply integer numbers.

You can switch representation of the ASNs to **asdot**, bear in mind however, that this **will influence the way regular expressions** are processed on the AS_PATH. To change the representation issue following command:

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R1(config)#as-format asdot
```

The output will immediately change:

```
RP/0/0/CPU0:SP1R1#sh ip bgp sum | i router
BGP router identifier 33.56.1.1, local AS number 10.10
```

To finish the changes, let's reconfigure SP1R2 to peer with newly configured AS 10.10:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 33.56.1.1
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#remote-as 10.10
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#ebgp-multihop 5
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
RP/0/0/CPU0:SP1R2#
```

The configuration of SP1R2 should look like this:

```
RP/0/0/CPU0:SP1R2#sh running-config router bgp
router bgp 3356
 bgp router-id 33.56.2.2
 address-family ipv4 unicast
  network 33.56.2.2/32
 !
 neighbor 33.56.1.1
  remote-as 10.10
  ebgp-multihop 5
  update-source Loopback0
  address-family ipv4 unicast
!
```

You'll immediately notice, that the BGP on IOS XR is protecting the eBGP session by default by requiring you to assign routing policy before announcing or accepting any prefixes on eBGP connections:

```
RP/0/0/CPU0:SP1R1#sh ip bgp sum
[…]

BGP is operating in STANDALONE mode.

Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker               3          3          3          3           3           0

Some configured eBGP neighbors (under default or non-default vrfs)
do not have both inbound and outbound policies configured for IPv4 Unicast
address family. These neighbors will default to sending and/or
receiving no routes and are marked with '!' in the output below.
Use the 'show bgp neighbor <nbr_address>' command for details.

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
33.56.2.2         0  3356      44      51        0    0    0 00:00:05           0!
```

As per description, information about mandatory eBGP route policy is also visible in neighbor output:

```
RP/0/0/CPU0:SP1R1#sh ip bgp nei 33.56.2.2
BGP neighbor is 33.56.2.2
 Remote AS 3356, local AS 10.10, external link
 Remote router ID 33.56.2.2
  BGP state = Established
  NSR State: None
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Last read 00:00:03, Last read before reset 00:00:00
[…]
 For Address Family: IPv4 Unicast
  BGP neighbor version 3
  Update group: 0.2 Filter-group: 0.0  No Refresh request being processed
  eBGP neighbor with no inbound or outbound policy; defaults to 'drop'
```



There's hack to that requirement – an '**bgp unsafe-ebgp-policy**' configured under VRF or BGP router level:

```
RP/0/0/CPU0:SP1R1(config)#router bgp 10.10
RP/0/0/CPU0:SP1R1(config-bgp)#bgp unsafe-ebgp-policy
```

It's sometimes useful for lab, but don't use this in production environment. The safeguard was created not without reason!
We'll learn how to properly define RPL policies in next task.



IOS XR also applies by default limits on the number of prefixes that can be accepted from a peer for a given address family. While with each version the limits are higher, it's worth to know in real world scenarios, that they may limit how many prefixes your IOS XR device will accept. For latest release in 6.1.x train, the current limits are:

- Number of neighbors – 4000 (can be extended to 15000)
- IPv4 Unicast: 1048576
- IPv4 Labeled-unicast: 131072
- IPv4 Tunnel: 1048576
- IPv6 Unicast: 524288
- IPv6 Labeled-unicast: 131072
- IPv4 Multicast: 131072
- IPv6 Multicast: 131072
- VPNv4 Unicast: 2097152
- IPv4 MDT: 131072
- VPNv6 Unicast: 1048576
- L2VPN EVPN: 2087152

The limits are documented in 'Implementing BGP' section of Configuration Guides for specific IOS XR release. One of the recent additions to IOS XR is command to verify the current scale of BGP:

```
RP/0/0/CPU0:SP2R3#show bgp scale

VRF: default
 Neighbors Configured: 10     Established: 10

 Address-Family Prefixes Paths    PathElem   Prefix  Path    PathElem
                                             Memory  Memory  Memory
  IPv4 Unicast  23       51       23         2.07KB  3.19KB  1.42KB
  IPv6 Unicast  22       58       22         2.23KB  3.63KB  1.40KB
  ------------------------------------------------------------------
  Total         45       109      45         4.30KB  6.81KB  2.81KB

Total VRFs Configured: 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.8: Filtering advertisements in IOS XR – using RPL

RPL is powerful language to express routing policies within IOS XR. We already built one simple route-policy using RPL that was modifying the BGP attributes on redistributing the networks from connected.

Now, we will demonstrate the filtering capabilities of RPL for AS_PATH attribute.

RPL uses two separate processors for AS_PATH – one comes directly from IOS and IOS-XE and is called **ios-regexp**. The other are set of built-in operators that are able to translate the regexp defined to both asdot and asplain formats without requiring any additional work:

```
RP/0/0/CPU0:SP1R1(config)#route-policy TEST
RP/0/0/CPU0:SP1R1(config-rpl)#if as-path ?
  in                Member of a set
  is-local          Route originates in this AS
  length            Length of BGP AS-path
  neighbor-is       BGP AS-path neighbor is
  originates-from   BGP AS-path originates-from
  passes-through    BGP AS-path passes-through
  unique-length     Length of BGP AS-path ignoring duplicate
```



ios-regexp behaves in the same way as it's IOS and IOS-XE prototype. This means, that for 2B/4B ASN matching in asdot format, you'll hit the same caveat as on those network operating systems – once the format on router is changed to asplain, regexp constructs trying to match dotted format of ASNs in AS_PATH will cease to match anything.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.8.1: Using RPL with ios-regexp to filter out the prefix while running asdot formatting

Let's build a sample route-map to filter incoming updates and check for regular expression of 10.10 (which will become "10\.10" to escape the special dot meaning). We will also build a 'pass-all' policy to satisfy the requirement to have both inbound and outbound policies configured and applied.

```
RP/0/0/CPU0:SP1R2(config)#route-policy RP-IOSREGEXP
RP/0/0/CPU0:SP1R2(config-rpl)#if as-path in ( ios-regex '10\.10' ) then pass endif
RP/0/0/CPU0:SP1R2(config)#route-policy RP-PASS
RP/0/0/CPU0:SP1R2(config-rpl)#pass
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
RP/0/0/CPU0:SP1R2(config)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes

RP/0/0/CPU0:SP1R2#sh running-config route-policy RP-IOSREGEXP
route-policy RM-IOSREGEXP
  if as-path in (ios-regex '10\.10') then
    pass
  endif
end-policy
!
```



route-police entries are processed sequentially just like route-map entries. They can also inherit one another, we'll see examples later on.

Now we will apply the policy inbound on updates from neighbor 33.56.1.1:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 33.56.1.1
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy RP-IOSREGEXP in
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy RP-PASS out
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#
```

To be able to test it, we need to also define policies on the neighbor – we'll use 'pass all' for now:

```
RP/0/0/CPU0:SP1R1(config)#route-policy RP-PASS
RP/0/0/CPU0:SP1R1(config-rpl)#pass
RP/0/0/CPU0:SP1R1(config-rpl)#end-policy
RP/0/0/CPU0:SP1R1(config)#router bgp 10.10
RP/0/0/CPU0:SP1R1(config-bgp)#neighbor 33.56.2.2
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#route-policy RP-PASS in
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#route-policy RP-PASS out
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#
```

After committing the configuration, SP1R1 sees the advertisement immediately:

```
RP/0/0/CPU0:SP1R1#sh ip bgp sum | i 33.56.2.2
33.56.2.2          0  3356     447     450       18    0    0 00:01:21          1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

And on the SP1R2 our policy to match '10.10' also worked for now:

```
RP/0/0/CPU0:SP1R2#sh ip bgp sum | i 33.56.1.1
33.56.1.1          0 10.10     510     516       63    0    0 00:02:07          1
RP/0/0/CPU0:SP1R2#sh ip bgp
[…]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 33.56.1.1/32       33.56.1.1                0             0 10.10 i
*> 33.56.2.2/32       0.0.0.0                  0         32768 i
```

Now, we'll change the SP1R2 setting to represent the AS format:

```
RP/0/0/CPU0:SP1R2(config)#no as-format asdot
RP/0/0/CPU0:SP1R2(config)#
RP/0/0/CPU0:SP1R2#clear ip bgp *
```

And then the prefix is gone, as the current regexp won't match anymore:

```
RP/0/0/CPU0:SP1R2#sh ip bgp
[…]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 33.56.2.2/32       0.0.0.0                  0         32768 i

Processed 1 prefixes, 1 paths

RP/0/0/CPU0:SP1R2#show bgp neighbor | i prefix
  0 accepted prefixes, 0 are bestpaths
  Cumulative no. of prefixes denied: 1.
  Maximum prefixes allowed 1048576
```

We'll change the route-policy relying on the ios-regexp module to IOS XR native operator:

```
RP/0/0/CPU0:SP1R2(config)#route-policy RP-BGP-ASN-PASS
RP/0/0/CPU0:SP1R2(config-rpl)#if as-path originates-from '10.10' then pass end-if
RP/0/0/CPU0:SP1R2(config-rpl)#exit
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 33.56.1.1
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy RP-BGP-ASN-PASS in
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

Now, the prefix originating from ASN 10.10 is accepted despite having asplain representation of 655370:

```
RP/0/0/CPU0:SP1R2#sh ip bgp sum
[…]

Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker             117        117        117        117         117           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
33.56.1.1         0 655370     523     531      117    0    0 00:06:07          1

RP/0/0/CPU0:SP1R2#sh ip bgp nei | i prefix
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  1 accepted prefixes, 1 are bestpaths
  Cumulative no. of prefixes denied: 1.
  Maximum prefixes allowed 1048576

RP/0/0/CPU0:SP1R2#sh ip bgp 33.56.1.1
BGP routing table entry for 33.56.1.1/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker               141         141
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  655370
    33.56.1.1 (metric 103) from 33.56.1.1 (33.56.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best, group-best
      Received Path ID 0, Local Path ID 0, version 141
      Origin-AS validity: not-found
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.8.2: Rollback SP1R1 to original configuration

Rollback the configuration to a last AS3356 configuration:

```
RP/0/0/CPU0:SP1R1#rollback configuration AS3356
Loading Rollback Changes.
Loaded Rollback Changes in 1 sec
Committing.
5 items committed in 1 sec (4)items/sec
Updating.
Updated Commit database in 1 sec
Configuration successfully rolled back commit 'AS3356'.

RP/0/0/CPU0:SP1R1#sh running-config router bgp
router bgp 3356
 bgp router-id 33.56.1.1
 address-family ipv4 unicast
  network 33.56.1.1/32
 !
  neighbor 33.56.2.2
  remote-as 3356
  update-source Loopback0
  address-family ipv4 unicast
  !
 !
 neighbor 33.56.3.3
  remote-as 3356
  update-source Loopback0
 !
 neighbor 33.56.5.5
  remote-as 3356
  update-source Loopback0
  address-family ipv4 unicast
  !
 !
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.9: BGP Address Families in IOS XR

As you already saw, IOS XR divides the configuration of BGP process to address families.

In general, we have an address family configuration per BGP process:

```
RP/0/0/CPU0:SP1R1#sh running-config router bgp
router bgp 3356
 bgp router-id 33.56.1.1
 address-family ipv4 unicast
  network 33.56.1.1/32
 !
 address-family ipv6 unicast
 !
```

And the neighbor configuration part, that activates (or not) specific AFs:

```
 neighbor 33.56.1.1
  remote-as 3356
  update-source Loopback0
  address-family ipv4 unicast
 !
 neighbor 33.56.3.3
  remote-as 3356
  update-source Loopback0
  address-family ipv4 unicast
 !
 neighbor 33.56.5.5
  remote-as 3356
  update-source Loopback0
  ! notice this neighbor is missing any AF assignment – it won't form any
  ! session unless some AF is configured here
 !
```

Task 1.9.1: IPv6 unicast session configuration in BGP

Now, we'll add IPv6 session and exchange the loopback IPv6. Configuration for SP1R2:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R2(config-bgp-af)#network 2001:33:56:aa::2/128
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 2001:33:56:aa::1
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#remote-as 3356
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#
```

And for SP1R1:

```
RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R1(config-bgp-af)#network 2001:33:56:aa::1/128
RP/0/0/CPU0:SP1R1(config-bgp)#neighbor 2001:33:56:aa::2
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#remote-as 3356
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#
```

Separate, IPv6 transport session was established and the prefixes were exchanged – view from SP1R1:

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R1#sh bgp ipv6 unicast summary
[…]

Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker               4          4          4          4           4           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
2001:33:56:aa::2
                  0  3356       4       4        4    0    0 00:00:50           1

RP/0/0/CPU0:SP1R1#sh bgp ipv6 unicast
[…]
   Network            Next Hop            Metric LocPrf Weight Path
*> 2001:33:56:aa::1/128
                      ::                       0         32768 i
*>i2001:33:56:aa::2/128
                      2001:33:56:aa::2
                                               0    100       0 i

Processed 2 prefixes, 2 paths
```

And from SP1R2:

```
RP/0/0/CPU0:SP1R2#sh bgp ipv6 unicast summary
[…]

Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker               4          4          4          4           4           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
2001:33:56:aa::1
                  0  3356       6       6        4    0    0 00:02:38           1

RP/0/0/CPU0:SP1R2#show bgp ipv6 unicast
[…]
   Network            Next Hop            Metric LocPrf Weight Path
*>i2001:33:56:aa::1/128
                      2001:33:56:aa::1
                                               0    100       0 i
*> 2001:33:56:aa::2/128
                      ::                       0         32768 i

Processed 2 prefixes, 2 paths
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.9.2: iBGP full mesh in AS3356

For the rest of the configuration in AS3356 (on SP1R3 and SP1R5) we'll use tools that speed up configuration of multiple similar sessions.

For IOS XR, there are three template groups you can use:

- **af-group** – grouping neighbors sharing specific AF
- **session-group** – grouping neighbors with specific session settings
- **neighbor-group** – grouping neighbors, that can share af-group and session-group by inheritance, but can also be full configuration

We'll use those tools to configure R1, R2 and R3. First, define the templates:

```
RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#af-group AF-IPV4 address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-afgrp)#exit
RP/0/0/CPU0:SP1R1(config-bgp)#af-group AF-IPV6 address-family ipv6 unicast
RP/0/0/CPU0:SP1R1(config-bgp-afgrp)#exit
RP/0/0/CPU0:SP1R1(config-bgp)#session-group SG-AS3356
RP/0/0/CPU0:SP1R1(config-bgp-sngrp)#update-source loopback 0
RP/0/0/CPU0:SP1R1(config-bgp-sngrp)#exit
RP/0/0/CPU0:SP1R1(config-bgp)#neighbor-group NG-AS3356-IPV4
RP/0/0/CPU0:SP1R1(config-bgp-nbrgrp)#tcp mss 1460
RP/0/0/CPU0:SP1R1(config-bgp-nbrgrp)#description AS3356 peer
RP/0/0/CPU0:SP1R1(config-bgp-nbrgrp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbrgrp-af)#use af-group AF-IPV4
RP/0/0/CPU0:SP1R1(config-bgp-nbrgrp-af)#exit
RP/0/0/CPU0:SP1R1(config-bgp-nbrgrp)#
RP/0/0/CPU0:SP1R1(config-bgp)#neighbor-group NG-AS3356-IPV6
RP/0/0/CPU0:SP1R1(config-bgp-nbrgrp)#tcp mss 1460
RP/0/0/CPU0:SP1R1(config-bgp-nbrgrp)#description AS3356 peer
RP/0/0/CPU0:SP1R1(config-bgp-nbrgrp)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbrgrp-af)#use af-group AF-IPV6
RP/0/0/CPU0:SP1R1(config-bgp-nbrgrp-af)#exit
RP/0/0/CPU0:SP1R1(config-bgp-nbrgrp)#
```



This is little extreme example of using all three options for illustration – while they could be compacted for example into one neighbor-group. In large configurations however, the flexibility is very important to enable scale and ease of configuration.

Now, apply the configuration templates to specific neighbors.

For SP1R1 the resulting configuration should look like this:

```
router bgp 3356
 bgp router-id 33.56.1.1
 address-family ipv4 unicast
  network 33.56.1.1/32
 !
 address-family ipv6 unicast
  network 2001:33:56:aa::1/128
 !
 af-group AF-IPV4 address-family ipv4 unicast
 !
 af-group AF-IPV6 address-family ipv6 unicast
 !
 session-group SG-AS3356
  update-source Loopback0
 !
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
 neighbor-group NG-AS3356-IPV4
  tcp mss 1460
  description AS3356 peer
  address-family ipv4 unicast
   use af-group AF-IPV4
  !
 !
 neighbor-group NG-AS3356-IPV6
  tcp mss 1460
  description AS3356 peer
  address-family ipv6 unicast
   use af-group AF-IPV6
  !
 !
 neighbor 33.56.2.2
  remote-as 3356
  use session-group SG-AS3356
  use neighbor-group NG-AS3356-IPV4
 !
 neighbor 33.56.3.3
  remote-as 3356
  use session-group SG-AS3356
  use neighbor-group NG-AS3356-IPV4
 !
 neighbor 33.56.5.5
  remote-as 3356
  use session-group SG-AS3356
  use neighbor-group NG-AS3356-IPV4
 !
 neighbor 2001:33:56:aa::2
  remote-as 3356
  use session-group SG-AS3356
  use neighbor-group NG-AS3356-IPV6
 !
 neighbor 2001:33:56:aa::3
  remote-as 3356
  use session-group SG-AS3356
  use neighbor-group NG-AS3356-IPV6
 !
 neighbor 2001:33:56:aa::5
  remote-as 3356
  use session-group SG-AS3356
  use neighbor-group NG-AS3356-IPV6
```

Apply the same template config and specific peer controls for other routers within AS3356 – SP1R2 and SP1R3.

In case of SP1R5 we'll use IOS-XE specific BGP configuration tools. IOS and IOS-XE offers peer session templates and peer policy templates. While they can't be used at the same time for one neighbor, they're also very useful. Session and policy templates are improved version of peer groups known from IOS for long time.

```
SP1-R5-RR(config)#router bgp 3356
SP1-R5-RR(config-router)#template peer-session PS-AS3356
SP1-R5-RR(config-router-stmp)#description AS3356 peers
SP1-R5-RR(config-router-stmp)#update-source loopback 0
SP1-R5-RR(config-router-stmp)#exit
SP1-R5-RR(config-router)#neighbor 33.56.1.1 remote-as 3356
SP1-R5-RR(config-router)#neighbor 33.56.1.1 inherit peer-session PS-AS3356
SP1-R5-RR(config-router)#neighbor 33.56.2.2 remote-as 3356
SP1-R5-RR(config-router)#neighbor 33.56.2.2 inherit peer-session PS-AS3356
SP1-R5-RR(config-router)#neighbor 33.56.3.3 remote-as 3356
SP1-R5-RR(config-router)#neighbor 33.56.3.3 inherit peer-session PS-AS3356
SP1-R5-RR(config-router)#address-family ipv4 unicast
SP1-R5-RR(config-router-af)#neighbor 33.56.1.1 activate
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
SP1-R5-RR(config-router-af)#neighbor 33.56.2.2 activate
SP1-R5-RR(config-router-af)#neighbor 33.56.3.3 activate
SP1-R5-RR(config-router-af)#network 33.56.5.5 mask 255.255.255.255
SP1-R5-RR(config-router-af)#address-family ipv6 unicast
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:aa::1 activate
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:aa::2 activate
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:aa::3 activate
SP1-R5-RR(config-router-af)#network 2001:33:56::5/128
```

Once the configuration is complete, it should look like this:

```
router bgp 3356
 template peer-session PS-AS3356
  description AS3356 peers
  update-source Loopback0
 exit-peer-session
 !
 bgp log-neighbor-changes
 neighbor 2001:33:56:AA::1 remote-as 3356
 neighbor 2001:33:56:AA::1 inherit peer-session PS-AS3356
 neighbor 2001:33:56:AA::2 remote-as 3356
 neighbor 2001:33:56:AA::2 inherit peer-session PS-AS3356
 neighbor 2001:33:56:AA::3 remote-as 3356
 neighbor 2001:33:56:AA::3 inherit peer-session PS-AS3356
 neighbor 33.56.1.1 remote-as 3356
 neighbor 33.56.1.1 inherit peer-session PS-AS3356
 neighbor 33.56.2.2 remote-as 3356
 neighbor 33.56.2.2 inherit peer-session PS-AS3356
 neighbor 33.56.3.3 remote-as 3356
 neighbor 33.56.3.3 inherit peer-session PS-AS3356
 !
 address-family ipv4
  network 33.56.5.5 mask 255.255.255.255
  no neighbor 2001:33:56:AA::1 activate
  no neighbor 2001:33:56:AA::2 activate
  no neighbor 2001:33:56:AA::3 activate
  neighbor 33.56.1.1 activate
  neighbor 33.56.2.2 activate
  neighbor 33.56.3.3 activate
 exit-address-family
 !
 address-family ipv6
  network 2001:33:56:AA::5/128
  neighbor 2001:33:56:AA::1 activate
  neighbor 2001:33:56:AA::2 activate
  neighbor 2001:33:56:AA::3 activate
 exit-address-family
```

If the configuration was done correctly, all routers within AS3356 should peer with each other over IPv4 and IPv6, exchanging one prefix on each transport session:

On SP1R1 (example, for IOS XR):

```
RP/0/0/CPU0:SP1R1#sh bgp ipv4 unicast summary | b Neigh
Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
33.56.2.2         0  3356      77       92       32    0    0 00:08:39          1
33.56.3.3         0  3356      52       58       32    0    0 00:08:44          1
33.56.5.5         0  3356      28       23       32    0    0 00:09:04          1
RP/0/0/CPU0:SP1R1#sh bgp ipv6 unicast summary | b Neigh
Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
2001:33:56:aa::2  0  3356      89       90       24    0    0 00:08:19          1
2001:33:56:aa::3
                  0  3356      52       58       24    0    0 00:08:46          1
2001:33:56:aa::5
                  0  3356      15       12       24    0    0 00:08:58          1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

On SP1R5:

```
SP1-R5-RR#sh ip bgp summary | begin Neig
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
33.56.1.1       4         3356      12      14       13    0    0 00:08:13        1
33.56.2.2       4         3356      11      13       13    0    0 00:07:48        1
33.56.3.3       4         3356      12      14       13    0    0 00:08:22        1
SP1-R5-RR#sh bgp ipv6 unicast summary | b Neighbor
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001:33:56:AA::1
                4         3356      11      14        7    0    0 00:07:59        1
2001:33:56:AA::2
                4         3356      11      14        7    0    0 00:07:43        1
2001:33:56:AA::3
                4         3356      12      14        7    0    0 00:08:35        1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.10: Configure SP1R4 and SP2R4 routers in AS 44.66

Configure IOS XR routers SP1R4 and SP2R4 for BGP routing in AS 44.66. The peering should be established between loopbacks.

To facilitate configuration for this lab, extend use of OSPFv2 and OSPFv3 to interfaces connecting SP1R4 and SP2R4, like this:

SP1R4:

```
router ospf as4466
 prefix-suppression
 address-family ipv4 unicast
 area 0
  interface Loopback0
   loopback stub-network enable
  !
  interface GigabitEthernet0/0/0/1
   network point-to-point
  !
 !
router ospfv3 as4466
 prefix-suppression
 redistribute connected
 area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/1
   network point-to-point
  !
 !
!
```

SP2R4:

```
router ospf as4466
 prefix-suppression
 address-family ipv4 unicast
 area 0
  interface Loopback0
   loopback stub-network enable
  !
  interface GigabitEthernet0/0/0/1
   network point-to-point
  !
 !
router ospfv3 as4466
 prefix-suppression
 redistribute connected
 area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/1
   network point-to-point
  !
 !
!
```

The end effect, as for AS3356 routers, is to have both IPv4 and IPv6 exchange loopback routes over BGP between SP1R4 and SP2R4:

```
RP/0/0/CPU0:SP1R4#sh ip bgp sum | b Neig
Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
12.39.4.4         0 44.66      16      16        4    0    0 00:12:50           1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R4#sh bgp ipv6 unicast summary | b Neig
Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
2001:12:39:ff::4
                  0 44.66       6       6        4    0    0 00:02:58          1

RP/0/0/CPU0:SP2R4#sh ip bgp sum | b Neig
Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
33.56.4.4         0 2883650      16      16        4    0    0 00:13:04          1

RP/0/0/CPU0:SP2R4#sh bgp ipv6 unicast summary | b Neig
Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
2001:33:56:aa::4
                  0 2883650       7       7        4    0    0 00:03:15          1
```



There's a subtle difference in the way IOS XR announces prefixes versus IOS, IOS-XE or some other vendor gear.

IOS XR checks for loop in the path BEFORE announcing prefix, so may not even announce prefixes that could cause loop in the other ASN.

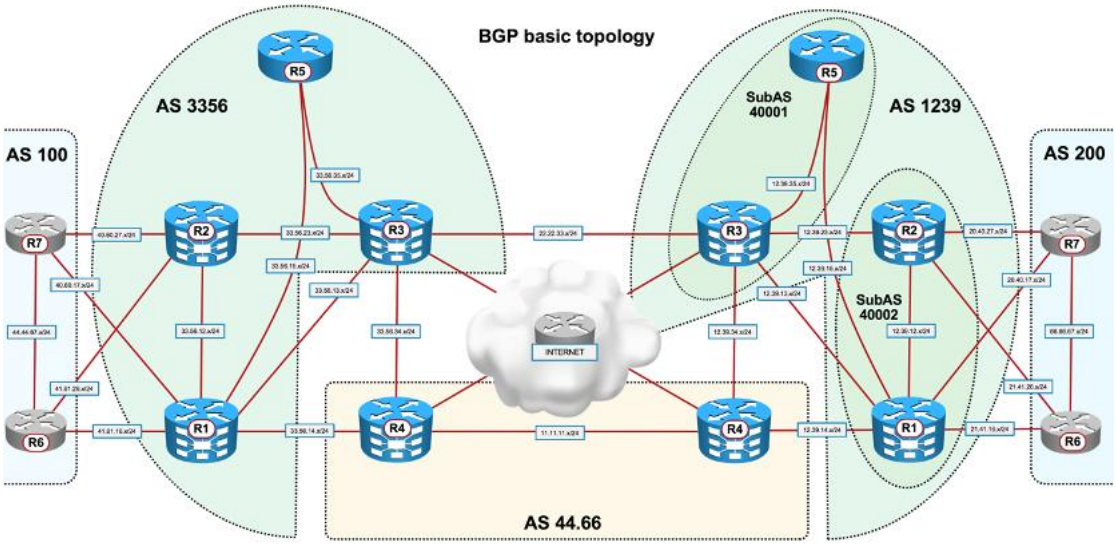
To change this behaviour and ignore the additional check, configure following command per AF:

```
as-path-loopcheck out disable
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.11: Configure confederation in AS 1239

Configure IOS XR routers R1, R2, R3 and IOS-XE router R5 to belong to AS 1239. SP2R3 and SP2R5 should belong to SubAS 40001, while SP2R1 and SP2R2 should belong to SubAS 40002, like on the topology below:



Confederations, both in IOS-XE and IOS XR are configured with the subAS confederation ASN as global one, with identifier pointing to true ASN number. For R3 and R5 in subAS 40001 belonging to AS1239 this would lead to following configuration:

```
RP/0/0/CPU0:SP2R3(config)#router bgp 40001
RP/0/0/CPU0:SP2R3(config-bgp)#bgp confederation identifier 1239
RP/0/0/CPU0:SP2R3(config-bgp)#bgp confederation peers
RP/0/0/CPU0:SP2R3(config-bgp-confed-peers)#40002
RP/0/0/CPU0:SP2R3(config-bgp-confed-peers)#exit
RP/0/0/CPU0:SP2R3(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R3(config-bgp-af)#network 12.39.3.3/32
RP/0/0/CPU0:SP2R3(config-bgp-af)#exit
RP/0/0/CPU0:SP2R3(config-bgp)#address-family ipv6 unicast
RP/0/0/CPU0:SP2R3(config-bgp-af)#network 2001:12:39:ff::3/128
RP/0/0/CPU0:SP2R3(config-bgp-af)#exit
RP/0/0/CPU0:SP2R3(config-bgp)#neighbor 2001:12:39:ff::5
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#remote-as 40001
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP2R3(config-bgp)#neighbor 12.39.5.5
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#remote-as 40001
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#commit
RP/0/0/CPU0:SP2R3(config-bgp)#
```

And for SP2R5:

```
SP2-R5-RR(config)#no router bgp 1239
SP2-R5-RR(config)#router bgp 40001
SP2-R5-RR(config-router)#bgp confederation identifier 1239
SP2-R5-RR(config-router)#bgp confederation peers 40002
SP2-R5-RR(config-router)#neighbor 12.39.3.3 remote-as 40001
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
SP2-R5-RR(config-router)#neighbor 12.39.3.3 update-source loo0
SP2-R5-RR(config-router)#neighbor 2001:12:39:ff::3 remote-as 40001
SP2-R5-RR(config-router)#neighbor 2001:12:39:ff::3 update-source loopback 0
SP2-R5-RR(config-router)#address-family ipv4 unicast
SP2-R5-RR(config-router-af)#neighbor 12.39.3.3 activate
SP2-R5-RR(config-router-af)#network 12.39.5.5 mask 255.255.255.255
SP2-R5-RR(config-router-af)#address-family ipv6 unicast
SP2-R5-RR(config-router-af)#neighbor 2001:12:39:FF::3 activate
SP2-R5-RR(config-router-af)#network 2001:12:39:ff::5/128
SP2-R5-RR(config-router-af)#^Z
```

The session should go up and again, exchange IPv4 and IPv6 prefix for Lo0 over separate transport sessions. Note, that AS identified is subAS we configured for the confederation:

```
RP/0/0/CPU0:SP2R3#sh bgp ipv4 unicast summary | b Nei
Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
12.39.5.5         0 40001      23      19        6    0    0 00:15:17          1

RP/0/0/CPU0:SP2R3#sh bgp ipv6 unicast summary | b Nei
Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
2001:12:39:ff::5
                  0 40001      20      16        4    0    0 00:12:37          1

SP2-R5-RR#sh bgp ipv4 unicast summary | b Nei
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
12.39.3.3       4        40001      19      24       10    0    0 00:16:01        1
SP2-R5-RR#sh bgp ipv6 unicast summary | b Nei
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001:12:39:FF::3
                4        40001      17      21        6    0    0 00:13:20        1
```

While the entry may loss competition to enter the BGP-learned Loopback0 route because IS-IS running in underlay is also advertising this route (and has lower administrative distance):

```
SP2-R5-RR#sh ip bgp
BGP table version is 10, local router ID is 12.39.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 r>i  12.39.3.3/32     12.39.3.3                0    100      0 i
 *>   12.39.5.5/32     0.0.0.0                  0         32768 i
```

…we can still ask BGP to show it's RIB and check that the route indeed is part of advertisement made within confederation:

```
SP2-R5-RR#sh ip bgp 12.39.3.3
BGP routing table entry for 12.39.3.3/32, version 3
Paths: (1 available, best #1, table default, RIB-failure(17))
  Not advertised to any peer
  Refresh Epoch 1
  Local
    12.39.3.3 (metric 10) from 12.39.3.3 (12.39.3.3)
      Origin IGP, metric 0, localpref 100, valid, confed-internal, best
      rx pathid: 0, tx pathid: 0x0
```

Now, we'll configure second subAS (40002) with SP2R1 and SP2R2, and then connect subASes together to form one AS 1239.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

For the confederation 40002, the resulting IOS XR configurations should look like this:

SP2R1:

```
router bgp 40002
 bgp confederation peers
  40001
 !
 bgp confederation identifier 1239
 address-family ipv4 unicast
  network 12.39.1.1/32
 !
 address-family ipv6 unicast
  network 2001:12:39:ff::1/128
 !
 neighbor 12.39.2.2
  remote-as 40002
  update-source Loopback0
  address-family ipv4 unicast
  !
 !
 neighbor 2001:12:39:ff::2
  remote-as 40002
  update-source Loopback0
  address-family ipv6 unicast
  !
 !
!
```

SP2R2:

```
router bgp 40002
 bgp confederation peers
  40001
 !
 bgp confederation identifier 1239
 address-family ipv4 unicast
  network 12.39.2.2/32
 !
 address-family ipv6 unicast
  network 2001:12:39:ff::2/128
 !
 neighbor 12.39.1.1
  remote-as 40002
  update-source Loopback0
  address-family ipv4 unicast
  !
 !
 neighbor 2001:12:39:ff::1
  remote-as 40002
  update-source Loopback0
  address-family ipv6 unicast
  !
 !
!
```

Connecting SP2R2 in 40002 to SP2R3 in 40001:

```
RP/0/0/CPU0:SP2R2(config)#router bgp 40002
RP/0/0/CPU0:SP2R2(config-bgp)#neighbor 12.39.3.3
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#remote-as 40001
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#address-family ipv4 unicast
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#ignore-connected-check
RP/0/0/CPU0:SP2R2(config-bgp)#neighbor 2001:12:39:ff::3
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#remote-as 40001
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#ignore-connected-check
```

And SP2R3 to connect to SP2R2:

```
RP/0/0/CPU0:SP2R3(config)#router bgp 40001
RP/0/0/CPU0:SP2R3(config-bgp)#neighbor 12.39.2.2
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#remote-as 40002
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#update-source lo0
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#ignore-connected-check
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#neighbor 2001:12:39:ff::2
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#remote-as 40002
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#update-source lo0
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#ignore-connected-check
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#commit
RP/0/0/CPU0:SP2R3#
```

SP2R3 to connect to SP2R1:

```
RP/0/0/CPU0:SP2R3(config)#router bgp 40001
RP/0/0/CPU0:SP2R3(config-bgp)#neighbor 12.39.1.1
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#remote-as 40002
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#ignore-connected-check
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#neighbor 2001:12:39:ff::1
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#remote-as 40002
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#ignore-connected-check
RP/0/0/CPU0:SP2R3(config-bgp-nbr)#address-family ipv6 unicast
```

And SP2R1 to connect to SP2R3 and SP2R5:

```
RP/0/0/CPU0:SP2R1(config)#router bgp 40002
RP/0/0/CPU0:SP2R1(config-bgp)#neighbor 12.39.3.3
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#remote-as 40001
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#update-source lo0
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#ignore-connected-check
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R1(config-bgp-nbr-af)#neighbor 2001:12:39:ff::3
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#remote-as 40001
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#update-source lo0
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#ignore-connected-check
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP2R1(config-bgp)#neighbor 12.39.5.5
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#remote-as 40001
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#ignore-connected-check
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R1(config-bgp-nbr-af)#neighbor 2001:12:39:ff::5
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#remote-as 40001
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#update-source loopback 0
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#ignore-connected-check
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#address-family ipv6 unicast
```

And finally, on SP2R5 to SP2R1:

```
SP2-R5-RR(config)#router bgp 40001
SP2-R5-RR(config-router)#neighbor 12.39.1.1 remote-as 40002
SP2-R5-RR(config-router)#neighbor 12.39.1.1 update-source loopback 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
SP2-R5-RR(config-router)#neighbor 12.39.1.1 disable-connected-check
SP2-R5-RR(config-router)#neighbor 2001:12:39:ff::1 remote-as 40002
SP2-R5-RR(config-router)#neighbor 2001:12:39:ff::1 update-source lo0
SP2-R5-RR(config-router)#address-family ipv4 unicast
SP2-R5-RR(config-router-af)#neighbor 12.39.1.1 activate
SP2-R5-RR(config-router-af)#address-family ipv6 unicast
SP2-R5-RR(config-router-af)#neighbor 2001:12:39:ff::1 activate
SP2-R5-RR(config-router-af)#commit
SP2-R5-RR#
```

End result is information about subASes within the scope of AS1239. Example from SP2R5:

```
SP2-R5-RR#sh bgp ipv4 un | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 r>   12.39.1.1/32     12.39.1.1                0    100      0 (40002) i
 r i                   12.39.1.1                0    100      0 (40002) i
 r>   12.39.2.2/32     12.39.2.2                0    100      0 (40002) i
 r i                   12.39.2.2                0    100      0 (40002) i
 r>i  12.39.3.3/32     12.39.3.3                0    100      0 i
 *>   12.39.5.5/32     0.0.0.0                  0         32768 i
SP2-R5-RR#sh bgp ipv6 un | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 r>   2001:12:39:FF::1/128
                       2001:12:39:FF::1
                                                0    100      0 (40002) i
 r i                   2001:12:39:FF::1
                                                0    100      0 (40002) i
 r>   2001:12:39:FF::2/128
                       2001:12:39:FF::2
                                                0    100      0 (40002) i
 r i                   2001:12:39:FF::2
                                                0    100      0 (40002) i
 r>i  2001:12:39:FF::3/128
                       2001:12:39:FF::3
                                                0    100      0 i
 *>   2001:12:39:FF::5/128
                        ::                      0         32768 i
```

As SP2R5 sees two paths to R1, we can also double check that both of paths are visible:

```
SP2-R5-RR#sh ip bgp 12.39.1.1
BGP routing table entry for 12.39.1.1/32, version 16
Paths: (2 available, best #1, table default, RIB-failure(17))
  Advertised to update-groups:
     1
  Refresh Epoch 1
  (40002)
    12.39.1.1 (metric 10) from 12.39.1.1 (12.39.1.1)
      Origin IGP, metric 0, localpref 100, valid, confed-external, best
      rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 1
  (40002)
    12.39.1.1 (metric 10) from 12.39.3.3 (12.39.3.3)
      Origin IGP, metric 0, localpref 100, valid, confed-internal
      rx pathid: 0, tx pathid: 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.12: Configure routers in AS 200

Configure IOS-XE routers R7 and R6 to peer over Loopback interfaces. Because there's no IGP configuration, please add OSPFv2 process 200 with Area 0 spanning Lo0 (passive) and Gi3, as well as OSPFv3 process 200 with the same parameters. Also, configure network point-to-point on Gi3 interfaces to save DR/BDR election.

Most important config snippets from R6:

CER6

```
interface Loopback0
 ip address 192.168.6.6 255.255.255.255
 ip ospf network point-to-point
 ip ospf 200 area 0
 ipv6 address 2001:22:44:EE::6/128
 ospfv3 200 ipv6 area 0
!
router ospfv3 200
 !
 address-family ipv6 unicast
 exit-address-family
!
router ospf 200
 passive-interface Loopback0
!
router bgp 200
 bgp router-id 192.168.6.6
 bgp log-neighbor-changes
 neighbor 2001:22:44:EE::7 remote-as 200
 neighbor 2001:22:44:EE::7 update-source Loopback0
 neighbor 192.168.7.7 remote-as 200
 neighbor 192.168.7.7 update-source Loopback0
 !
 address-family ipv4
  network 192.168.6.6 mask 255.255.255.255
  redistribute connected
  no neighbor 2001:22:44:EE::7 activate
  neighbor 192.168.7.7 activate
 exit-address-family
 !
 address-family ipv6
  redistribute connected
  network 2001:22:44:EE::6/128
  neighbor 2001:22:44:EE::7 activate
 exit-address-family
```

Most important config snippets from R7:

CER7

```
interface Loopback0
 ip address 192.168.7.7 255.255.255.255
 ip ospf network point-to-point
 ip ospf 200 area 0
 ipv6 address 2001:22:44:EE::7/128
 ospfv3 200 ipv6 area 0
!
router ospfv3 200
 !
 address-family ipv6 unicast
 exit-address-family
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
router ospf 200
 passive-interface Loopback0
!
router bgp 200
 bgp router-id 192.168.7.7
 bgp log-neighbor-changes
 neighbor 2001:22:44:EE::6 remote-as 200
 neighbor 2001:22:44:EE::6 update-source Loopback0
 neighbor 192.168.6.6 remote-as 200
 neighbor 192.168.6.6 update-source Loopback0
 !
 address-family ipv4
  network 192.168.6.6 mask 255.255.255.255
  redistribute connected
  no neighbor 2001:22:44:EE::6 activate
  neighbor 192.168.6.6 activate
 exit-address-family
 !
 address-family ipv6
  redistribute connected
  network 2001:22:44:EE::6/128
  neighbor 2001:22:44:EE::6 activate
 exit-address-family
!
!
```

End effect on R6:

```
CER6#sh bgp ipv4 unicast summary | b Ne
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.7.7     4          200      11      11       10    0    0 00:04:54        4
CER6#sh bgp ipv6 unicast summary | b Ne
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001:22:44:EE::7
                4          200      12      12       22    0    0 00:02:37        5
```

End effect on R7:

```
CER7#sh bgp ipv4 uni summ | b Ne
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.6.6     4          200      12      12       18    0    0 00:05:47        5
CER7#sh bgp ipv6 uni summ | b Ne
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001:22:44:EE::6
                4          200      13      13       22    0    0 00:03:30        5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: eBGP peering

Task 2.1: Configure peering between AS100 and AS3356

Configure eBGP session between R6 and SP1R1 using physical interface addresses.

Configure eBGP session between R7 and SP1R2 using physical interface addresses.

Shut down link between CEL7 and SP1R1.

Shut down link between CEL6 and SP1R2.

Shut down link between CEL7 and SP2R3.

Shut down link between CEL6 and SP2R4.

Please do not make any additional policy configuration on eBGP sessions apart from required ‘pass’ for inbound and outbound in case of IOS XR sessions.

Task 2.2: Configure peering between AS3356 and AS44.66

Configure eBGP session between SP1R3 and SP1R4.

Configure eBGP session between SP1R1 and SP1R4.

Shut down link between SP1R2 and SP1R4.

Please do not make any additional policy configuration on eBGP sessions apart from required ‘pass’ for inbound and outbound in case of IOS XR sessions.

Task 2.3: Configure peering between AS3356 and AS1239

Configure eBGP session between SP1R3 and SP2R3.

Please do not make any additional policy configuration on eBGP sessions apart from required ‘pass’ for inbound and outbound in case of IOS XR sessions.

Task 2.4: Configure peering between AS1239 and AS44.66

Configure eBGP session between SP2R3 and SP2R4.

Configure eBGP session between SP2R4 and SP2R1.

Shut down link between SP2R2 and SP2R4.

Please do not make any additional policy configuration on eBGP sessions apart from required ‘pass’ for inbound and outbound in case of IOS XR sessions.

Task 2.5: Configure peering between AS1239 and AS200

Configure eBGP session between CER6 and SP2R1.

Configure eBGP session between CER7 and SP2R2.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Shut down link between CER7 and SP1R1.

Shut down link between CER6 and SP1R2.

Shut down link between SP1R3 and CER7.

Shut down link between SP1R4 and CER6.

Please do not make any additional policy configuration on eBGP sessions apart from required ‘pass’ for inbound and outbound in case of IOS XR sessions.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 5: BGP traffic engineering (TE)

For the BGP traffic engineering we'll use clean config, with IGP and BGP configured in last chapter.

Please pay attention to topology drawings and reconfiguration tasks.

This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch05.virl1.virl`
- **For VIRL²/CML²:** `sp.bootcamp.ch05.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch05.unl`

Task 1: Inter-AS Traffic engineering

As you already probably noticed, some prefixes are not reachable past the first edge BGP router. This happens because on iBGP sessions within AS, BGP doesn't by default change next-hop, and between some of the ASes we have prefixes that are not advertised broadly, so BGP can't find valid next-hop and doesn't install the route in routing table.

We'll fix that and meet other requirements below in this task.

Task 1.1: AS100 Traffic Engineering

Your task will be to make sure AS100 routers use SP1R2 router as ingress for all traffic bound to AS3356.

For all other traffic, SP1R1 should be used.

First of all, let's check current visibility from AS100 point of view. For CEL6:

```
CEL6#sh ip bgp | b Netw
     Network          Next Hop            Metric LocPrf Weight Path
 *>   12.39.4.4/32     41.61.16.1                             0 3356 2883650 i
 *>   33.56.1.1/32     41.61.16.1               0             0 3356 i
 * i                   40.60.27.2               0    100      0 3356 i
 *>   33.56.2.2/32     41.61.16.1                             0 3356 i
 * i                   40.60.27.2               0    100      0 3356 i
 *>   33.56.3.3/32     41.61.16.1                             0 3356 i
 * i                   40.60.27.2               0    100      0 3356 i
 *>   33.56.4.4/32     41.61.16.1                             0 3356 2883650 i
 *>   33.56.5.5/32     41.61.16.1                             0 3356 i
 * i                   40.60.27.2               0    100      0 3356 i
 *>i  40.60.27.0/24    172.16.7.7               0    100      0 i
 *>   41.61.16.0/24    0.0.0.0                  0         32768 i
```

Quite naturally, CEL6 sees remote prefixes received over eBGP from SP1R1 (his direct peer) as better than the ones passed over from CEL7. In this case it's because point #7 of BGP path selection - eBGP sessions are preferred over iBGP ones.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński



Short version of BGP path selection:

1. Highest WEIGHT
2. Highest LOCAL_PREF
3. Prefer locally originated prefix via ‘network’, ‘aggregate’ or redistribution
4. Shortest AS_PATH
5. Lowest origin type (IGP < EGP < INCOMPLETE)
6. Lowest MED
7. Prefer eBGP over iBGP
8. Prefer path with lower IGP metric to the BGP next-hop
9. Determine if multipath is being used
10. When both paths are external, prefer the older
11. Prefer the route received from BGP speaker with lower router ID
12. If router ID (or originator ID) is the same for multiple paths, prefer the path with minimum cluster list length
13. Prefer the path that comes from the lowest neighbor address

There’s also a knob to let BGP stay with the path it selected previously if the router-id is the same: “**bgp bestpath compare-routerid**”.

To satisfy requirement of the task, we should:

Make sure CEL7 is the best path to all prefixes originated in AS3356

Make sure CEL6 is the best path for all other prefixes

The easiest way will be to set local preference to something higher than default 100, matching specific prefixes on both routers. We also need to remember to properly reset next-hop on both routers for prefixes advertised to inside of AS100.

On CEL6 let’s configure route-map that will set LOCAL_PREF to 200 to ALL prefixes that use AS 3356 as transit:

```
CEL6(config)#ip as-path access-list 100 permit ^3356_([0-9]+)+$

! this translates to: match any prefixes that were advertised from 3356 and have
! one or more ASNs in the path after that

CEL6(config)#route-map BGP-AS3356 permit 10
CEL6(config-route-map)#match as-path 100
CEL6(config-route-map)#set local-preference 200
CEL6(config-route-map)#route-map BGP-AS3356 permit 20

! we need to permit all other prefixes – otherwise they'd be blocked

CEL6(config)#router bgp 100
CEL6(config-router)#address-family ipv4 unicast
CEL6(config-router-af)#network 172.16.6.6 mask 255.255.255.255
CEL6(config-router-af)#network 44.44.67.0 mask 255.255.255.0
CEL6(config-router-af)#neighbor 172.16.7.7 next-hop-self
CEL6(config-router-af)#neighbor 41.61.16.1 route-map BGP-AS3356 in
CEL6(config-router-af)#address-family ipv6 unicast
CEL6(config-router-af)#neighbor  2001:44:66:CC::7 next-hop-self
CEL6(config-router-af)#neighbor 2001:44:66:CC::7 route-map BGP-AS3356 in
CEL6(config-router-af)#network 2001:44:66:CC::6/128
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CEL6(config-router-af)#network 2001:44:44:67::/64
CEL6(config-route-map)#exit
```

Right now, we can see that the policy works – the prefixes sent from behind of AS3356 are properly matched and their LOCAL_PREF is adjusted to 200:

```
CEL6#sh ip bgp summary | b Nei
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
41.61.16.1      4         3356      24      14       15    0    0 00:04:05        6
172.16.7.7      4          100      11      11       15    0    0 00:04:13        5

CEL6#sh ip bgp | b Netw
     Network          Next Hop            Metric LocPrf Weight Path
 *>   12.39.4.4/32     41.61.16.1                        200      0 3356 2883650 i
 *>   33.56.1.1/32     41.61.16.1               0             0 3356 i
 * i                   172.16.7.7               0    100      0 3356 i
 *>   33.56.2.2/32     41.61.16.1                             0 3356 i
 * i                   172.16.7.7               0    100      0 3356 i
 *>   33.56.3.3/32     41.61.16.1                             0 3356 i
 * i                   172.16.7.7               0    100      0 3356 i
 *>   33.56.4.4/32     41.61.16.1                        200      0 3356 2883650 i
 *>   33.56.5.5/32     41.61.16.1                             0 3356 i
 * i                   172.16.7.7               0    100      0 3356 i
 *>i  40.60.27.0/24    172.16.7.7               0    100      0 i
 *>   41.61.16.0/24    0.0.0.0                  0         32768 i
```

On CEL7 let's configure route-map that will set LOCAL_PREF to 200, and leave other prefixes as is:

```
CEL7(config)#ip as-path access-list 100 permit ^3356$
CEL7(config)#route-map BGP-AS3356 permit 10
CEL7(config-route-map)#match as-path 100
CEL7(config-route-map)#set local-preference 200
CEL7(config-route-map)#route-map BGP-AS3356 permit 20
CEL7(config-route-map)#exit
CEL7(config)#router bgp 100
CEL7(config-router)#address-family ipv4 unicast
CEL7(config-router-af)#neighbor 172.16.6.6 next-hop-self
CEL7(config-router-af)#neighbor 40.60.27.2 route-map BGP-AS3356 in
CEL7(config-router-af)#network 172.16.7.7 mask 255.255.255.255
CEL6(config-router-af)#network 44.44.67.0 mask 255.255.255.0
CEL7(config-router-af)#address-family ipv6 unicast
CEL7(config-router-af)#neighbor 2001:44:66:CC::6 next-hop-self
CEL7(config-router-af)#neighbor 2001:40:60:27::2 route-map BGP-AS3356 in
CEL7(config-router-af)#network 2001:44:66:CC::7/128
CEL6(config-router-af)#network 2001:44:44:67::/64
```

And now let's clear BGP sessions and check the result:

```
CEL7#sh ip bgp | b Netw
     Network          Next Hop            Metric LocPrf Weight Path
 *>i  12.39.4.4/32     172.16.6.6               0    100      0 3356 2883650 i
 *>   33.56.1.1/32     40.60.27.2                        200      0 3356 i
 *>   33.56.2.2/32     40.60.27.2               0    200      0 3356 i
 *>   33.56.3.3/32     40.60.27.2                        200      0 3356 i
 r>i  33.56.4.4/32     172.16.6.6               0    100      0 3356 2883650 i
 *>   33.56.5.5/32     40.60.27.2                        200      0 3356 i
 *>   40.60.27.0/24    0.0.0.0                  0         32768 i
 *>i  41.61.16.0/24    172.16.6.6               0    100      0 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2: AS3356 to AS100 Traffic Engineering

Your task will be to make sure AS3356 routers use SP1R2 as egress for all traffic bound to AS100 prefixes. Traffic to AS100 loopback space (172.16/16) should however transit the nearest exit point from AS3356 to AS100.

Again, we need to make sure we reset next-hop at the edge of our AS border and make sure the loopback address space is assigned some special value. Because we're not asked to provide the same policy within all of AS3356, let's set weight which is local to router – we'll only touch LOCAL_PREF on SP1R2 to make sure it attracts traffic to AS100:

On SP1R1:

```
RP/0/0/CPU0:SP1R1(config)#route-policy From-AS100
RP/0/0/CPU0:SP1R1(config-rpl)#if destination in ( 172.16.0.0/16 le 32 ) then
RP/0/0/CPU0:SP1R1(config-rpl-if)#set weight 100
RP/0/0/CPU0:SP1R1(config-rpl-if)#else
RP/0/0/CPU0:SP1R1(config-rpl-else)#pass
RP/0/0/CPU0:SP1R1(config-rpl-else)#endif
RP/0/0/CPU0:SP1R1(config-rpl)#exit

RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#neighbor 41.61.16.6
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#route-policy From-AS100 in
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#neighbor 2001:41:61:16::6
RP/0/0/CPU0:SP1R1(config-bgp-nbr)# address-family ipv6 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#route-policy From-AS100 in

! we will also make sure to modify next-hop – either per neighbor
! or in af-group

RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#neighbor 33.56.2.2
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#next-hop-self
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#neighbor 33.56.3.3
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#next-hop-self
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#neighbor 33.56.5.5
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#next-hop-self
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#neighbor 2001:33:56:aa::2
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#next-hop-self
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#neighbor 2001:33:56:aa::3
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#next-hop-self
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#neighbor 2001:33:56:aa::5
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#next-hop-self

Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes

RP/0/0/CPU0:SP1R1#clear ip bgp 41.61.16.6
RP/0/0/CPU0:SP1R1#clear ip bgp 2001:41:61:16::6
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński



IOS XR has built-in text editor for some of the features, including route-policies. edit policy inline using one of three editors, call it from exec mode like this:

```
RP/0/0/CPU0:SP1R1#edit route-policy From-AS100 ?
  emacs  to use Emacs editor
  nano   to use nano editor
  vim    to use Vim editor
  <cr>
```

After refreshing sessions we can see that the policy was applied correctly:

```
RP/0/0/CPU0:SP1R1#sh ip bgp | b Netw
  Network            Next Hop            Metric LocPrf Weight Path
* i12.39.1.1/32      22.22.33.254                100      0 1239 i
* i12.39.2.2/32      22.22.33.254                100      0 1239 i
* i12.39.3.3/32      22.22.33.254             0  100      0 1239 i
* i12.39.4.4/32      33.56.34.4                  100      0 44.66 i
*>                   33.56.14.4                           0 44.66 i
* i12.39.5.5/32      22.22.33.254                100      0 1239 i
*> 33.56.1.1/32      0.0.0.0                  0         32768 i
*>i33.56.2.2/32      33.56.2.2                0  100      0 i
*>i33.56.3.3/32      33.56.3.3                0  100      0 i
* i33.56.4.4/32      33.56.34.4               0  100      0 44.66 i
*>                   33.56.14.4               0           0 44.66 i
*>i33.56.5.5/32      33.56.5.5                0  100      0 i
* i40.60.27.0/24     40.60.27.7               0  100      0 100 i
*>                   41.61.16.6                           0 100 i
* i41.61.16.0/24     40.60.27.7                  100      0 100 i
*>                   41.61.16.6               0           0 100 i
*> 172.16.6.6/32     41.61.16.6               0          100 100 i
* i172.16.7.7/32     33.56.2.2                0  100      0 100 i
*>                   41.61.16.6                          100 100 i
```

On SP1R2 we have to do similar thing, this time matching not the prefix itself, but origin AS (loopbacks will be included, and thus – adjusted):

```
RP/0/0/CPU0:SP1R2(config)#route-policy From-AS100
RP/0/0/CPU0:SP1R2(config-rpl)#if as-path originates-from '100' then
RP/0/0/CPU0:SP1R2(config-rpl-if)#set weight 100

 ! local to the router

RP/0/0/CPU0:SP1R2(config-rpl-if)#set local-preference 200

 ! to make sure we're default gateway to AS100 from all of AS3356

RP/0/0/CPU0:SP1R2(config-rpl-if)#else
RP/0/0/CPU0:SP1R2(config-rpl-else)#pass
RP/0/0/CPU0:SP1R2(config-rpl-else)#endif
RP/0/0/CPU0:SP1R2(config-rpl)#exit
```

And then for adjusting the policies and next-hops:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 33.56.1.1
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#next-hop-self
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#neighbor 33.56.3.3
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#next-hop-self
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#neighbor 33.56.5.5
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#next-hop-self
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#neighbor 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy From-AS100 in
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#neighbor 2001:33:56:aa::1
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#next-hop-self
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#neighbor 2001:33:56:aa::3
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#next-hop-self
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#neighbor 2001:33:56:aa::5
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#next-hop-self
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#neighbor 2001:40:60:27::7
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy From-AS100 in
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
RP/0/0/CPU0:SP1R2#clear ip bgp *
```

Now, on SP1R2:

```
RP/0/0/CPU0:SP1R2#sh ip bgp | b Netw
Network            Next Hop            Metric LocPrf Weight Path
* i12.39.1.1/32      22.22.33.254                100      0 1239 i
* i12.39.2.2/32      22.22.33.254                100      0 1239 i
* i12.39.3.3/32      22.22.33.254             0  100      0 1239 i
*>i12.39.4.4/32      33.56.1.1                   100      0 2883650 i
* i                  33.56.34.4                  100      0 2883650 i
* i12.39.5.5/32      22.22.33.254                100      0 1239 i
*>i33.56.1.1/32      33.56.1.1                0  100      0 i
*> 33.56.2.2/32      0.0.0.0                  0         32768 i
*>i33.56.3.3/32      33.56.3.3                0  100      0 i
*>i33.56.4.4/32      33.56.1.1                0  100      0 2883650 i
* i                  33.56.34.4               0  100      0 2883650 i
*>i33.56.5.5/32      33.56.5.5                0  100      0 i
*> 40.60.27.0/24     40.60.27.7               0  200      0 100 i
*> 41.61.16.0/24     40.60.27.7                  200      0 100 i
*> 44.44.67.0/24     40.60.27.7               0  200      0 100 i
*>i172.16.6.6/32     33.56.1.1                0  200      0 100 i
* i172.16.7.7/32     33.56.1.1                   200      0 100 i
*>                   40.60.27.7               0  200      0 100 i
```

Notice, that for 172.16.6.6 we're not receiving second path from R7. Why's that? Let's check:

```
CEL7#sh ip bgp neighbors 40.60.27.2 advertised-routes | b Netw

     Network          Next Hop            Metric LocPrf Weight Path
 *>i  12.39.4.4/32     172.16.6.6               0    200      0 3356 2883650 i
 r>i  33.56.4.4/32     172.16.6.6               0    200      0 3356 2883650 i
 *>   40.60.27.0/24    0.0.0.0                  0         32768 i
 *>i  41.61.16.0/24    172.16.6.6               0    100      0 i
 *>   44.44.67.0/24    0.0.0.0                  0         32768 i
 r>i  172.16.6.6/32    172.16.6.6               0    100      0 i
 *>   172.16.7.7/32    0.0.0.0                  0         32768 i
```

The problem we're hitting is that SP1R2 on receiving the prefix from CEL7 does recursive lookup for next-hop and finds it's his own:

```
RP/0/0/CPU0:SP1R2#debug bgp update 40.60.27.7
RP/0/0/CPU0:SP1R2#clear log
Clear logging buffer [confirm with only 'y' or 'n'] [y/n] :y
RP/0/0/CPU0:SP1R2#clear ip bgp 40.60.27.7
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
 […wait for BGP convergence…]
RP/0/0/CPU0:SP1R2#sh log
RP/0/0/CPU0:SP1R2#
12:16:30.089 : bgp[1052]: [default-rtr] (ip4u): Received unreachables from 40.60.27.7:
attrcode=0, attrwdrflags=0x00000001
12:16:30.089 : bgp[1052]: [default-rtr] (ip4u): UPDATE from 40.60.27.7 with attributes:
12:16:30.089 : bgp[1052]: [default-rtr] (ip4u): nexthop 40.60.27.2/32, origin i, path
100
12:16:30.089 : bgp[1052]: [default-rtr] (ip4u): UPDATE from 40.60.27.7, prefix
172.16.6.6/32 (path ID: none) DENIED due to:
12:16:30.089 : bgp[1052]: [default-rtr] (ip4u):  next-hop is our own address;
RP/0/0/CPU0:SP1R2#undebug all
```

Actually, we can check this discrepancy between BGP view of the world with that of underlying OSPF instance on CEL7:

```
CEL7#sh ip cef 172.16.6.6 detail
172.16.6.6/32, epoch 2
  1 RR source [no flags]
  nexthop 40.60.27.2 GigabitEthernet2
CEL7#sh ip route 172.16.6.6
Routing entry for 172.16.6.6/32
  Known via "ospf 1", distance 110, metric 4, type intra area
  Last update from 40.60.27.2 on GigabitEthernet2, 01:06:50 ago
  Routing Descriptor Blocks:
  * 40.60.27.2, from 172.16.6.6, 01:06:50 ago, via GigabitEthernet2
      Route metric is 4, traffic share count is 1
```

So 172.16.6.6 is received via OSPF from SP1R2! This happens because CEL7 and CEL6 belong to the OSPF area 2, and their reachability information needs to be exchanged over area 0 – which SP1R1 and SP1R2 are members of.

Because it's a test lab, let's drop the OSPF and OSPFv3 adjacency over "eBGP" links between the ASes:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#router ospf core
RP/0/0/CPU0:SP1R2(config-ospf)#area 2
RP/0/0/CPU0:SP1R2(config-ospf-ar)#no int gi0/0/0/1
RP/0/0/CPU0:SP1R2(config-ospf-ar)#router ospfv3 core
RP/0/0/CPU0:SP1R2(config-ospfv3)#area 2
RP/0/0/CPU0:SP1R2(config-ospfv3-ar)#no int gi0/0/0/1
RP/0/0/CPU0:SP1R2(config-ospfv3-ar)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

And on SP1R1:

```
RP/0/0/CPU0:SP1R1(config)#router ospf core
RP/0/0/CPU0:SP1R1(config-ospf)#area 2
RP/0/0/CPU0:SP1R1(config-ospf-ar)#no interface GigabitEthernet0/0/0/1
RP/0/0/CPU0:SP1R1(config-ospf-ar)#router ospfv3 core
RP/0/0/CPU0:SP1R1(config-ospfv3)#area 2
RP/0/0/CPU0:SP1R1(config-ospfv3-ar)#no interface GigabitEthernet0/0/0/1
RP/0/0/CPU0:SP1R1(config-ospfv3-ar)#
```

Now, the situation looks like it should – on SP1R2:

```
RP/0/0/CPU0:SP1R2#sh ip bgp | b Netw
   Network            Next Hop            Metric LocPrf Weight Path
* i12.39.1.1/32      22.22.33.254                100      0 1239 i
* i12.39.2.2/32      22.22.33.254                100      0 1239 i
* i12.39.3.3/32      22.22.33.254             0  100      0 1239 i
*>i12.39.4.4/32      33.56.1.1                   100      0 2883650 i
* i                  33.56.34.4                  100      0 2883650 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
* i12.39.5.5/32       22.22.33.254                 100      0 1239 i
*>i33.56.1.1/32       33.56.1.1                0   100      0 i
*> 33.56.2.2/32       0.0.0.0                  0        32768 i
*>i33.56.3.3/32       33.56.3.3                0   100      0 i
*>i33.56.4.4/32       33.56.1.1                0   100      0 2883650 i
* i                    33.56.34.4               0   100      0 2883650 i
*>i33.56.5.5/32       33.56.5.5                0   100      0 i
*> 40.60.27.0/24      40.60.27.7               0   200    100 100 i
*> 41.61.16.0/24      40.60.27.7                   200    100 100 i
*> 44.44.67.0/24      40.60.27.7               0   200    100 100 i
* i172.16.6.6/32      33.56.1.1                0   100      0 100 i
*>                    40.60.27.7                   200    100 100 i
* i172.16.7.7/32      33.56.1.1                    100      0 100 i
*>                    40.60.27.7               0   200    100 100 i
```

And on SP1R1:

```
RP/0/0/CPU0:SP1R1#sh ip bgp | b Netw
Network            Next Hop            Metric LocPrf Weight Path
* i12.39.1.1/32       22.22.33.254                 100      0 1239 i
* i12.39.2.2/32       22.22.33.254                 100      0 1239 i
* i12.39.3.3/32       22.22.33.254             0   100      0 1239 i
* i12.39.4.4/32       33.56.34.4                   100      0 44.66 i
*>                    33.56.14.4                            0 44.66 i
* i12.39.5.5/32       22.22.33.254                 100      0 1239 i
*> 33.56.1.1/32       0.0.0.0                  0        32768 i
*>i33.56.2.2/32       33.56.2.2                0   100      0 i
*>i33.56.3.3/32       33.56.3.3                0   100      0 i
* i33.56.4.4/32       33.56.34.4               0   100      0 44.66 i
*>                    33.56.14.4               0            0 44.66 i
*>i33.56.5.5/32       33.56.5.5                0   100      0 i
*>i40.60.27.0/24      33.56.2.2                0   200      0 100 i
*                     41.61.16.6                            0 100 i
*>i41.61.16.0/24      33.56.2.2                    200      0 100 i
*                     41.61.16.6               0            0 100 i
*>i44.44.67.0/24      33.56.2.2                0   200      0 100 i
*                     41.61.16.6               0            0 100 i
* i172.16.6.6/32      33.56.2.2                    200      0 100 i
*>                    41.61.16.6               0          100 100 i
* i172.16.7.7/32      33.56.2.2                0   200      0 100 i
*>                    41.61.16.6                         100 100 i
```

We can check this also on SP1R5 just to make sure:

```
SP1-R5-RR#sh ip route bgp
[…]
      12.0.0.0/32 is subnetted, 1 subnets
B        12.39.4.4 [200/0] via 33.56.1.1, 02:35:56
      40.0.0.0/24 is subnetted, 1 subnets
B        40.60.27.0 [200/0] via 33.56.2.2, 00:04:57
      41.0.0.0/24 is subnetted, 1 subnets
B        41.61.16.0 [200/0] via 33.56.2.2, 00:04:57
      44.0.0.0/24 is subnetted, 1 subnets
B        44.44.67.0 [200/0] via 33.56.2.2, 00:04:57
      172.16.0.0/32 is subnetted, 2 subnets
B        172.16.6.6 [200/0] via 33.56.2.2, 00:04:57
B        172.16.7.7 [200/0] via 33.56.2.2, 00:04:57
```

And BGP view on SP1R5:

```
SP1-R5-RR#sh ip bgp | grep "100 i"
 *>i  40.60.27.0/24    33.56.2.2                0    200      0 100 i
 *>i  41.61.16.0/24    33.56.2.2                     200      0 100 i
 *>i  44.44.67.0/24    33.56.2.2                0    200      0 100 i
 *>i  172.16.6.6/32    33.56.2.2                     200      0 100 i
 * i                   33.56.1.1                0    100      0 100 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
*>i  172.16.7.7/32     33.56.2.2                0    200      0 100 i
* i                     33.56.1.1                     100      0 100 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.3: Announcing aggregates in IOS XR and in IOS-XE

In current configuration, AS3356 is announcing each loopback from 33.56.0.0/16 space as original /32:

```
CEL7#sh ip route bgp | i 33.56
B        33.56.1.1 [20/0] via 40.60.27.2, 1d10h
B        33.56.2.2 [20/0] via 40.60.27.2, 1d10h
B        33.56.3.3 [20/0] via 40.60.27.2, 1d08h
B        33.56.4.4 [200/0] via 172.16.6.6, 04:58:02
B        33.56.5.5 [20/0] via 40.60.27.2, 1d07h
```

To make it more optimal, we can aggregate the network advertisements and configure summary routes on both SP1R1 and SP1R2:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-af)#aggregate-address 33.56.0.0/16 summary-only
RP/0/0/CPU0:SP1R2(config-bgp-af)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes

RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-af)#aggregate-address 33.56.0.0/16 summary-only
RP/0/0/CPU0:SP1R1(config-bgp-af)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
RP/0/0/CPU0:SP1R1#
```

Both CEL6 and CEL7 will now only see one prefix coming for 33.56.0.0 network – the aggregated 33.56.0.0/16:

```
CEL7#sh ip route bgp | i 33.56
B        33.56.0.0 [20/0] via 40.60.27.2, 00:01:27
CEL6#sh ip route bgp | i 33.56
B        33.56.0.0 [200/0] via 172.16.7.7, 00:02:11
```

Let's configure the same aggregation for CEL6/CEL7 loopback space (172.16.0.0/24) towards AS3356:

```
RP/0/0/CPU0:SP1R1#sh route ipv4 bgp | i 172.16
B    172.16.6.6/32 [20/900] via 41.61.16.6, 05:03:04
B    172.16.7.7/32 [20/900] via 41.61.16.6, 05:03:04

RP/0/0/CPU0:SP1R2#sh route ipv4 bgp | i 172.16
B    172.16.6.6/32 [20/0] via 40.60.27.7, 1d09h
B    172.16.7.7/32 [20/0] via 40.60.27.7, 1d09h
```

Now, on CEL6:

```
CEL6(config)#router bgp 100
CEL6(config-router)#address-family ipv4 unicast
CEL6(config-router-af)#aggregate-address 172.16.0.0 255.255.0.0 summary-only
```

And on CEL7:

```
CEL7(config)#router bgp 100
CEL7(config-router)#address-family ipv4 unicast
CEL7(config-router-af)#aggregate-address 172.16.0.0 255.255.0.0 summary-only
```

Now, on SP1R1 we can see only the freshly installed aggregated summary:

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R1#sh route ipv4 bgp | i 172.16
B    172.16.0.0/16 [20/900] via 41.61.16.6, 00:02:42
```

And on SP1R2:

```
RP/0/0/CPU0:SP1R2#sh route ipv4 bgp | i 172.16
B    172.16.0.0/16 [20/0] via 40.60.27.7, 00:01:42
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.4: AS3356 to AS44.66 Traffic Engineering

Your task will be to make sure AS3356 routers use SP1R1 as egress for all traffic bound to AS44.66 prefixes.

Please remember about setting next-hop-self on iBGP session on AS-border routers.

Also, make sure to AS_PATH prepend all advertisements from AS3356 towards AS44.66 when advertising any path originated in AS100 like this:

```
RP/0/0/CPU0:SP1R1(config)#route-policy BGP-PREPEND
RP/0/0/CPU0:SP1R1(config-rpl)#if as-path originates-from '100' then
RP/0/0/CPU0:SP1R1(config-rpl-if)#prepend as-path most-recent 4
RP/0/0/CPU0:SP1R1(config-rpl-if)#else
RP/0/0/CPU0:SP1R1(config-rpl-else)#pass
RP/0/0/CPU0:SP1R1(config-rpl-else)#endif
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.5: AS3356 and AS1239 Traffic Engineering

Those ASes have only one link, but to make sure you have end-to-end connectivity within your pod, please remember about setting next-hop-self on iBGP session on AS-border routers.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.6: AS1239 to AS200 Traffic Engineering

AS1239 and AS200 agreed to work together in terms of network traffic engineering. Both ASNs agreed that traffic bound to AS200 will use SP2R2 as preferred egress point, and that MED can be used and will be honored to satisfy this requirement.

As MED influences the preference of path used, either AS200 needs to send the metric of the prefixes to AS1239, or AS1239 needs to set it up on receiving the prefixes from AS200.

We'll configure first AS200 to send MED (metric) information towards AS1239 routers, and then AS1239 routers to make a decision based on the metric attribute. As you remember, the lower the MED value, the more preferred the entry point to AS is.

```
CER6(config)#route-map RM-BGP-MED permit 10
CER6(config-route-map)#set metric ?
  +/-<metric>     Add or subtract metric
  <0-4294967295>  Metric value or Bandwidth in Kbits per second
CER6(config-route-map)#set metric 200
CER6(config)#router bgp 200
CER6(config-router)#add ipv4 unicast
CER6(config-router-af)#neighbor 21.41.16.1 route-map RM-BGP-MED out
CER6(config-router-af)#add ipv6 unicast
CER6(config-router-af)#neighbor 2001:21:41:16::1 route-map RM-BGP-MED out
```

We'll configure first AS200 to send MED (metric) information towards AS1239 routers, and then AS1239 routers to make a decision based on the metric attribute.

```
CER7(config)#route-map RM-BGP-MED permit 10
CER7(config-route-map)#set metric 50
CER7(config-route-map)#exit
CER7(config)#router bgp 200
CER7(config-router)#address-family ipv4 unicast
CER7(config-router-af)#neighbor 20.40.27.2 route-map RM-BGP-MED out
CER7(config-router-af)#address-family ipv6 unicast
CER7(config-router-af)#neighbor 2001:20:40:27::2 route-map RM-BGP-MED out
```

On SP2R1 and SP2R2 routers please remember to apply 'next-hop-self' to iBGP peerings within AS1239 before configuring anything.

After applying the configurations, on SP2R1 you can see following BGP information:

```
RP/0/0/CPU0:SP2R1#sh ip bgp | utility egrep "200 [i\?]"
*>i20.40.27.0/24      12.39.2.2                50    100      0 200 ?
*                     21.41.16.6              200             0 200 ?
*>i21.41.16.0/24      12.39.2.2                50    100      0 200 ?
*                     21.41.16.6              200             0 200 ?
*>i66.66.67.0/24      12.39.2.2                50    100      0 200 ?
*                     21.41.16.6              200             0 200 ?
*>i192.168.6.6/32     12.39.2.2                50    100      0 200 i
*                     21.41.16.6              200             0 200 i
*>i192.168.7.7/32     12.39.2.2                50    100      0 200 ?
*                     21.41.16.6              200             0 200 ?
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

And on SP2R2:

```
RP/0/0/CPU0:SP2R2#sh ip bgp | utility egrep "200 [i\?]"
*> 20.40.27.0/24      20.40.27.7               50             0 200 ?
*> 21.41.16.0/24      20.40.27.7               50             0 200 ?
*> 21.41.26.0/24      20.40.27.7               50             0 200 ?
*> 41.61.46.0/24      20.40.27.7               50             0 200 ?
*> 42.62.37.0/24      20.40.27.7               50             0 200 ?
*> 66.66.67.0/24      20.40.27.7               50             0 200 ?
*> 192.168.6.6/32     20.40.27.7               50             0 200 i
*> 192.168.7.7/32     20.40.27.7               50             0 200 ?
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.7: AS200 to AS1239 Traffic Engineering

Apart from agreements mentioned in Task 4.4, AS200 engineers wants to send traffic originated in AS100 over SP2R1 router, and traffic originated in AS3356 over SP2R2 router.

Prefixes from AS100 are marked at the SP1R3 with community of 100:100.

Prefixes from AS3356 are marked at the SP1R3 with community of 100:3356.

To mark the prefixes with specific communities, we need first to configure route-policy on SP1R3:

```
RP/0/0/CPU0:SP1R3(config)#route-policy MarkAtEdge
RP/0/0/CPU0:SP1R3(config-rpl)#if as-path originates-from '100' then
RP/0/0/CPU0:SP1R3(config-rpl-if)#set community (100:100)
RP/0/0/CPU0:SP1R3(config-rpl-if)#elseif as-path is-local then
RP/0/0/CPU0:SP1R3(config-rpl-elseif)#set community (100:3356)
RP/0/0/CPU0:SP1R3(config-rpl-elseif)#endif
RP/0/0/CPU0:SP1R2(config-rpl)#pass
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
RP/0/0/CPU0:SP1R3(config)#commit
RP/0/0/CPU0:SP1R3(config)#router bgp 3356
RP/0/0/CPU0:SP1R3(config-bgp)#neighbor 22.22.33.254
RP/0/0/CPU0:SP1R3(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R3(config-bgp-nbr-af)#route-policy MarkAtEdge in
RP/0/0/CPU0:SP1R3(config-bgp-nbr-af)#send-community-ebgp
RP/0/0/CPU0:SP1R3(config-bgp)#neighbor 2001:22:22:33::254
RP/0/0/CPU0:SP1R3(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R3(config-bgp-nbr-af)#send-community-ebgp
RP/0/0/CPU0:SP1R3(config-bgp-nbr-af)#route-policy MarkAtEdge in
RP/0/0/CPU0:SP1R3(config-bgp-nbr-af)#commit
```



IOS XR by default sends communities only to iBGP peers.

To send communities to eBGP peers use '**send-community-ebgp**' under specific AF. To send extended communities to eBGP peers use '**send-extended-community-ebgp**'.

Then make sure SP2R1 and SP2R2 propagate the communities to AS200 peers:

On SP2R1:

```
RP/0/0/CPU0:SP2R1(config)#router bgp 40002
RP/0/0/CPU0:SP2R1(config-bgp)#neighbor 21.41.16.6
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R1(config-bgp-nbr-af)#send-community-ebgp
RP/0/0/CPU0:SP2R1(config-bgp-nbr-af)#neighbor 2001:21:41:16::6
RP/0/0/CPU0:SP2R1(config-bgp-nbr)# address-family ipv6 unicast
RP/0/0/CPU0:SP2R1(config-bgp-nbr-af)#send-community-ebgp
RP/0/0/CPU0:SP2R1(config-bgp-nbr-af)#commit
```

And on SP2R2:

```
RP/0/0/CPU0:SP2R2(config)#router bgp 40002
RP/0/0/CPU0:SP2R2(config-bgp)#neighbor 20.40.27.7
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R2(config-bgp-nbr-af)#send-community-ebgp
RP/0/0/CPU0:SP2R2(config-bgp-nbr-af)#neighbor 2001:20:40:27::7
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP2R2(config-bgp-nbr-af)#send-community-ebgp
RP/0/0/CPU0:SP2R2(config-bgp-nbr-af)#commit
```

On CER6 and CER7 the prefixes should be visible with proper community tags. It's worth to note, that by default IOS and IOS-XE merge the community values to one integer, and they need '**ip bgp new-format**' command to show community values separated by colon:

```
CER6#sh ip bgp 172.16.6.6 | i Commu
Community: 6553700
CER6#conf t
CER6(config)#ip bgp new-format
CER6#sh ip bgp 172.16.6.6 | i Commu
      Community: 100:100
```

Now, we can configure the policies to match community and properly manipulate the LOCAL_PREF.

On CER7:

```
CER7(config)#ip community-list 20 permit 100:3356
CER7(config)#route-map RM-BGP-COMM permit 10
CER7(config-route-map)#match community 20
CER7(config-route-map)#set local-preference 2020
CER7(config)#route-map RM-BGP-COMM permit 20
CER7(config-route-map)#exit
CER7(config)#router bgp 200
CER7(config-router)#address-family ipv4 unicast
CER7(config-router-af)#neighbor 20.40.27.2 route-map RM-BGP-COMM in
CER7(config-router-af)#address-family ipv6 unicast
CER7(config-router-af)#neighbor 2001:20:40:27::2 route-map RM-BGP-COMM in
```

On CER6:

```
CER6(config)#ip community-list 10 permit 100:100
CER6(config)#route-map RM-BGP-COMM permit 10
CER6(config-route-map)#match community 10
CER6(config-route-map)#set local-preference 1010
CER6(config-route-map)#route-map RM-BGP-COMM permit 20
CER6(config-route-map)#exit
CER6(config)#router bgp 200
CER6(config-router)#address-family ipv4 unicast
CER6(config-router-af)#neighbor 21.41.16.1 route-map RM-BGP-COMM in
CER6(config-router-af)#address-family ipv6 unicast
CER6(config-router-af)#neighbor 2001:21:41:16::1 route-map RM-BGP-COMM in
```

This should give us what we were looking for. On CER7:

```
CER7#sh ip bgp | b Netw
     Network          Next Hop            Metric LocPrf Weight Path
 *>   12.39.1.1/32     20.40.27.2                             0 1239 i
 * i                   192.168.6.6              0    100      0 1239 i
 *>   12.39.2.2/32     20.40.27.2               0             0 1239 i
 * i                   192.168.6.6              0    100      0 1239 i
 *>   12.39.3.3/32     20.40.27.2                             0 1239 i
 * i                   192.168.6.6              0    100      0 1239 i
 *>   12.39.4.4/32     20.40.27.2                             0 1239 2883650 i
 * i                   192.168.6.6              0    100      0 1239 2883650 i
 *>   12.39.5.5/32     20.40.27.2                             0 1239 i
 * i                   192.168.6.6              0    100      0 1239 i
 *>   20.40.27.0/24    0.0.0.0                  0         32768 ?
 *>i  21.41.16.0/24    192.168.6.6              0    100      0 ?
 *>i  21.41.26.0/24    192.168.6.6              0    100      0 ?
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
 *>   33.56.1.1/32     20.40.27.2                    2020      0 1239 3356 i
 *>   33.56.2.2/32     20.40.27.2                    2020      0 1239 3356 i
 *>   33.56.3.3/32     20.40.27.2                    2020      0 1239 3356 i
 *>   33.56.4.4/32     20.40.27.2                              0 1239 2883650 i
 * i                   192.168.6.6              0    100       0 1239 2883650 i
 *>   33.56.5.5/32     20.40.27.2                    2020      0 1239 3356 i
 *    40.60.27.0/24    20.40.27.2                              0 1239 3356 100 i
 *>i                   192.168.6.6              0   1010       0 1239 3356 100 i
 *    41.61.16.0/24    20.40.27.2                              0 1239 3356 100 i
 *>i                   192.168.6.6              0   1010       0 1239 3356 100 i
 *>i  41.61.46.0/24    192.168.6.6              0    100       0 ?
 *>   42.62.37.0/24    0.0.0.0                  0         32768 ?
 *    44.44.67.0/24    20.40.27.2                              0 1239 3356 100 i
 *>i                   192.168.6.6              0   1010       0 1239 3356 100 i
 *>   66.66.67.0/24    0.0.0.0                  0         32768 ?
 * i                   192.168.6.6              0    100       0 ?
 * i  172.16.0.0       192.168.6.6              0    100       0 1239 3356 100 i
 *>                   20.40.27.2                              0 1239 3356 100 i
 *>   192.168.6.6/32   66.66.67.6               2         32768 i
 * i                   192.168.6.6              0    100       0 i
 *>   192.168.7.7/32   0.0.0.0                  0         32768 ?
```

On CER6:

```
CER6#sh ip bgp | b Ne
     Network          Next Hop            Metric LocPrf Weight Path
 * i  12.39.1.1/32     192.168.7.7              0    100       0 1239 i
 *>                   21.41.16.1               0              0 1239 i
 * i  12.39.2.2/32     192.168.7.7              0    100       0 1239 i
 *>                   21.41.16.1                             0 1239 i
 * i  12.39.3.3/32     192.168.7.7              0    100       0 1239 i
 *>                   21.41.16.1                             0 1239 i
 * i  12.39.4.4/32     192.168.7.7              0    100       0 1239 2883650 i
 *>                   21.41.16.1                             0 1239 2883650 i
 * i  12.39.5.5/32     192.168.7.7              0    100       0 1239 i
 *>                   21.41.16.1                             0 1239 i
 *>i  20.40.27.0/24    192.168.7.7              0    100       0 ?
 *>   21.41.16.0/24    0.0.0.0                  0         32768 ?
 *>   21.41.26.0/24    0.0.0.0                  0         32768 ?
 *>i  33.56.1.1/32     192.168.7.7              0   2020       0 1239 3356 i
 *                    21.41.16.1                             0 1239 3356 i
 *>i  33.56.2.2/32     192.168.7.7              0   2020       0 1239 3356 i
 *                    21.41.16.1                             0 1239 3356 i
 *>i  33.56.3.3/32     192.168.7.7              0   2020       0 1239 3356 i
 *                    21.41.16.1                             0 1239 3356 i
 * i  33.56.4.4/32     192.168.7.7              0    100       0 1239 2883650 i
 *>                   21.41.16.1                             0 1239 2883650 i
 *>i  33.56.5.5/32     192.168.7.7              0   2020       0 1239 3356 i
 *                    21.41.16.1                             0 1239 3356 i
 *>   40.60.27.0/24    21.41.16.1                    1010      0 1239 3356 100 i
 *>   41.61.16.0/24    21.41.16.1                    1010      0 1239 3356 100 i
 *>   41.61.46.0/24    0.0.0.0                  0         32768 ?
 *>i  42.62.37.0/24    192.168.7.7              0    100       0 ?
 *>   44.44.67.0/24    21.41.16.1                    1010      0 1239 3356 100 i
 * i  66.66.67.0/24    192.168.7.7              0    100       0 ?
 *>                   0.0.0.0                  0         32768 ?
 *>   172.16.0.0       21.41.16.1                              0 1239 3356 100 i
 * i                   192.168.7.7              0    100       0 1239 3356 100 i
 * i  192.168.6.6/32   192.168.7.7              2    100       0 i
 *>                   0.0.0.0                  0         32768 i
 r>i  192.168.7.7/32   192.168.7.7              0    100       0 ?
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4: Intra-AS Traffic engineering

Task 4.1: AS3356 Traffic Engineering (with route reflector)

AS3356 topology is now fully meshed with iBGP. It means that every BGP router within AS3356 has optimal view of the AS, but unfortunately it's also configuration-consuming to change any policy for the whole AS3356.

We will migrate the AS3356 to use BGP Route Reflector feature. SP1R5 will be RR, and all other routers will peer only with SP1R5 to get full view of the BGP network.

On SP1R5:

```
SP1-R5-RR(config)#router bgp 3356
SP1-R5-RR(config-router)#address-family ipv4 unicast
SP1-R5-RR(config-router-af)#neighbor 33.56.1.1 route-reflector-client
SP1-R5-RR(config-router-af)#neighbor 33.56.2.2 route-reflector-client
SP1-R5-RR(config-router-af)#neighbor 33.56.3.3 route-reflector-client
SP1-R5-RR(config-router)#address-family ipv6 unicast
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:AA::1 route-reflector-client
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:AA::2 route-reflector-client
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:AA::3 route-reflector-client
SP1-R5-RR(config-router-af)#^Z
```

Now, we can deconfigure any iBGP sessions configured on SP1R1, SP1R2 and SP1R3 apart from that leading to SP1R5:

On SP1R1:

```
RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#no neighbor 33.56.2.2
RP/0/0/CPU0:SP1R1(config-bgp)#no neighbor 33.56.3.3
RP/0/0/CPU0:SP1R1(config-bgp)#no neighbor 2001:33:56:aa::2
RP/0/0/CPU0:SP1R1(config-bgp)#no neighbor 2001:33:56:aa::3
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

On SP1R2:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#no neighbor 33.56.1.1
RP/0/0/CPU0:SP1R2(config-bgp)#no neighbor 33.56.3.3
RP/0/0/CPU0:SP1R2(config-bgp)#no neighbor 2001:33:56:aa::1
RP/0/0/CPU0:SP1R2(config-bgp)#no neighbor 2001:33:56:aa::3
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

On SP1R3:

```
RP/0/0/CPU0:SP1R3#conf t
RP/0/0/CPU0:SP1R3(config)#router bgp 3356
RP/0/0/CPU0:SP1R3(config-bgp)#no neighbor 33.56.1.1
RP/0/0/CPU0:SP1R3(config-bgp)#no neighbor 33.56.2.2
RP/0/0/CPU0:SP1R3(config-bgp)#no neighbor 2001:33:56:aa::1
RP/0/0/CPU0:SP1R3(config-bgp)#no neighbor 2001:33:56:aa::2
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

Comparing this to the BGP view before the introduction of Route Reflector we can see that as per design, only best paths are reflected towards Route Reflector Clients, and routers SP1R1, SP1R2 and SP1R3 lost redundant reachability information in BGP.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

In the Advanced BGP class we will focus on that problem.



If you'll need to apply any policies on the BGP RRs, there's special way to configure router to enable enforcing those changes. By default and according to best practices, you shouldn't need normally to modify any attributes on the RR = they should Reflect not Modify, right?

For IOS and IOS-XE routers, the correct method is to do:

```
neighbor X.X.X.X allow-policy
```

For IOS XR routers, you need to add globally for BGP process following statement:

```
ibgp policy out enforce-modifications
```

As soon as this is done, route policies attached to this specific neighbor definition will become effective.

For IOS-XE there's different knob to enable changing of next-hop:

```
neighbor X.X.X.X next-hop-self all
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.2: AS1239 Traffic Engineering

Within BGP Confederation BGP speakers in different subAS use eBGP sessions, but attributes important from Traffic Engineering point of view are retained – next hop, MED and local preference in particular.

To be able to do TE at the /32 and /128 level, please remove summary-only aggregates from the AS 3356 and AS 1239.

Let's take a look at the 33.56.2.2 prefix visible from SP2R3:

```
RP/0/0/CPU0:SP2R3#sh ip bgp 33.56.2.2
BGP routing table entry for 33.56.2.2/32
Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):
    0.1 0.4
  Advertised to peers (in unique update groups):
    12.39.5.5
  Path #1: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.1 0.4
  Advertised to peers (in unique update groups):
    12.39.5.5
  3356
    22.22.33.1 from 22.22.33.1 (33.56.3.3)
      Origin IGP, localpref 100, valid, external, best, group-best
      Received Path ID 0, Local Path ID 0, version 173
      Community: 100:3356
      Origin-AS validity: not-found
```

And now the same prefix from different subAS – 40002, but the same AS – 1239 – from SP2R2:

```
RP/0/0/CPU0:SP2R3#sh ip bgp 33.56.2.2
BGP routing table entry for 33.56.2.2/32
Paths: (2 available, best #1)
  Advertised to peers (in unique update groups):
    20.40.27.7      12.39.3.3
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    20.40.27.7      12.39.3.3
  (40001) 3356
    12.39.1.1 (metric 10) from 12.39.1.1 (12.39.1.1)
      Origin IGP, localpref 100, valid, confed-internal, best, group-best
      Received Path ID 0, Local Path ID 0, version 142
      Community: 100:3356
  Path #2: Received by speaker 0
  Not advertised to any peer
  (40001) 3356
    12.39.3.3 (metric 10) from 12.39.3.3 (12.39.3.3)
      Origin IGP, localpref 100, valid, confed-external
      Received Path ID 0, Local Path ID 0, version 0
      Community: 100:3356
```

While the attributes are preserved, we can see that SP2R2 choose 12.39.1.1 as best way to reach 33.56.2.2 which may not make much sense given it's additional hop on the way. However, given the costs to both Loopbacks are the same (10), BGP had to choose the lowest Router-ID from both advertisements.

Let's modify the cost of going through the SP2R2<>SP2R1 interface:

```
RP/0/0/CPU0:SP2R2(config)#router isis core
RP/0/0/CPU0:SP2R2(config-isis)#interface gigabitEthernet 0/0/0/2
RP/0/0/CPU0:SP2R2(config-isis-if)#address-family ipv4 unicast
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP2R2(config-isis-if-af)#metric 30
RP/0/0/CPU0:SP2R2(config-isis-if-af)#
```

We can see that the preferred path was immediately changed to that leading directly to SP2R3:

```
RP/0/0/CPU0:SP2R2#sh ip bgp 33.56.2.2
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    12.39.1.1       20.40.27.7
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    12.39.1.1       20.40.27.7
  (40001) 3356
    12.39.3.3 (metric 10) from 12.39.3.3 (12.39.3.3)
      Origin IGP, localpref 100, valid, confed-external, best, group-best
      Received Path ID 0, Local Path ID 0, version 174
      Community: 100:3356
```

Please also note, we didn't change the IPv6 IS-IS metric, which means the topology for IS-IS is still using path via SP1R1 (2001:12:39:ff::1):

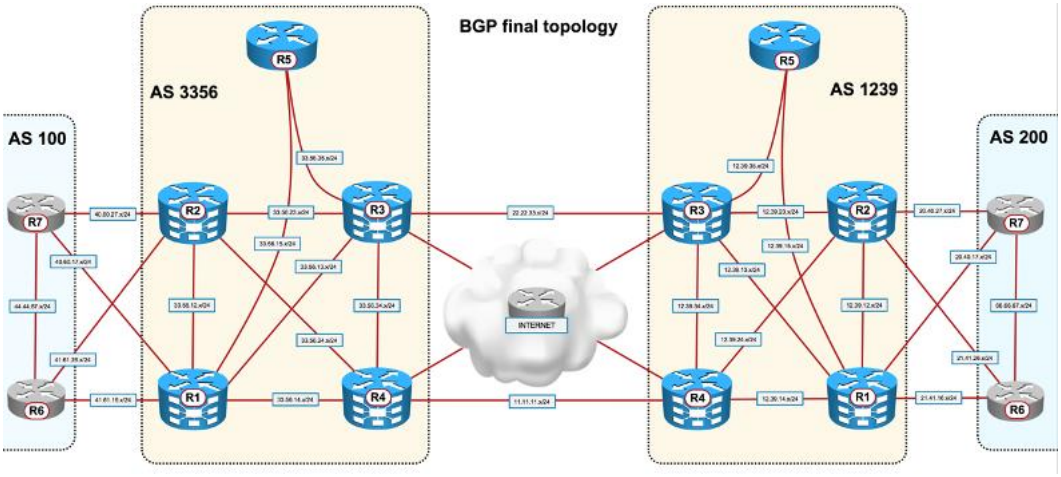
```
RP/0/0/CPU0:SP2R2#sh bgp ipv6 unicast 2001:33:56:aa::2
BGP routing table entry for 2001:33:56:aa::2/128
Paths: (2 available, best #1)
  Advertised to peers (in unique update groups):
    2001:20:40:27::7
    2001:12:39:ff::3
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    2001:20:40:27::7
    2001:12:39:ff::3
  (40001) 3356
    2001:12:39:ff::1 (metric 10) from 2001:12:39:ff::1 (12.39.1.1)
      Origin IGP, localpref 100, valid, confed-internal, best, group-best
      Received Path ID 0, Local Path ID 0, version 293
      Community: 100:3356
  Path #2: Received by speaker 0
  Not advertised to any peer
  (40001) 3356
    2001:12:39:ff::3 (metric 10) from 2001:12:39:ff::3 (12.39.3.3)
      Origin IGP, localpref 100, valid, confed-external
      Received Path ID 0, Local Path ID 0, version 0
      Community: 100:3356
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5: Wrap-up

Please stop the current simulation and close the lab.

Now, load clean config from the beginning of the session – **SP CCIE bootcamp clean config - week 3.unl**. Your end goal will be to make your pod reach below state, with R5 routers in both AS1239 and AS3356 acting as RRs for those ASes, and AS100 and AS200 remaining connected redundantly to their upstream ASes:



Task 5.1: AS100 setup

Configure AS100 between CER6 and CER7.

Peerings should be between Loopback0 interfaces and addresses of the routers.

Reconfigure IGP: please use OSPFv2 (for IPv4) and OSPFv3 (for IPv6) with process numbers 100 running within the AS. Do not add the links between AS100 and AS3356 to OSPF!

Make sure to protect the peering with password AS100.

Task 5.2: AS3356

Configure AS3356 as containing SP1R1, SP1R2, SP1R3, SP1R4 and SP1R5 with SP1R5 acting as BGP Route Reflector for IPv4 and IPv6 traffic.

Peerings should be between Loopback0 interfaces and addresses of the routers.

Reconfigure IGP: please use OSPFv2 (for IPv4) and OSPFv3 (for IPv6) with process name AS3356 running within the AS. Do not add the links between AS100 and AS3356 or between AS3356 and AS1239 to OSPF!

Make sure to protect the peering with password AS3356.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5.3: AS1239

Configure AS1239 as containing SP2R1, SP2R2, SP2R3, SP2R4 and SP2R5 with SP2R5 acting as BGP Route Reflector for IPv4 and IPv6 traffic.

Peerings should be between Loopback0 interfaces and addresses of the routers.

Reconfigure IGP: please use ISIS with process name AS1239 running within the AS. Do not add the links between AS200 and AS1239 or between AS3356 and AS1239 to ISIS!

Make sure to protect the peering with password AS1239.

Task 5.4: AS200

Configure AS200 between CER6 and CER7.

Peerings should be between Loopback0 interfaces and addresses of the routers.

Reconfigure IGP: please use OSPFv2 (for IPv4) and OSPFv3 (for IPv6) with process numbers 200 running within the AS. Do not add the links between AS200 and AS1239 to OSPF!

Make sure to protect the peering with password AS200.

Task 5.5: Inter-AS BGP peerings

Peer AS100 with AS3356, configuring sessions between R7 and R2 and between R6 and R1. Use password LEFT.

Peer AS3356 with AS1239, configuring sessions between R3 routers in both ASes and between R4 routers in both ASes. Use password CENTER.

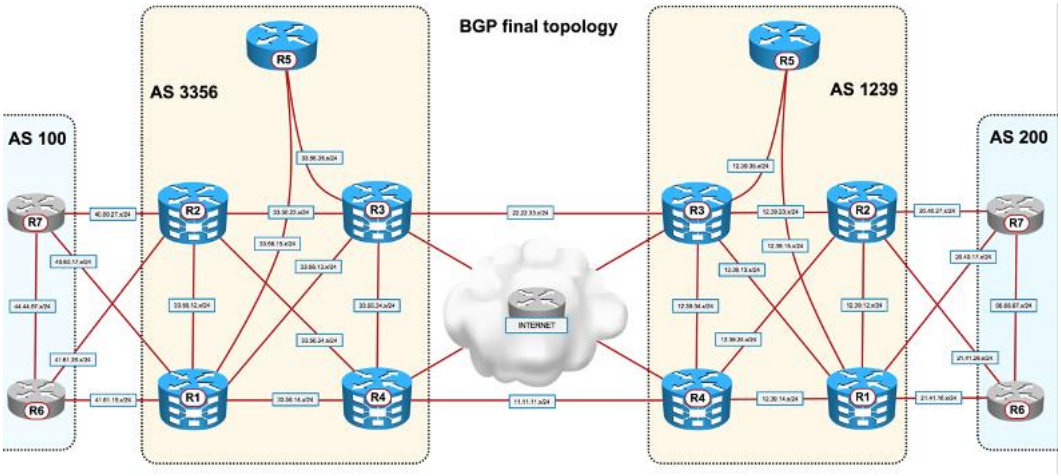
Peer AS1239 with AS200, configuring sessions between R2 and R7 and between R1 and R6 routers. Use password RIGHT.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 6: Advanced BGP

For BGP advanced part we'll use the topology built during the last task (#5) of BGP basic lab.

Your topology and logical configuration should reflect this picture:



This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch06.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch06.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch06.unl`



Please note that in the interest of clear representation of specific technologies and protocols, in most of the cases IPv4 is used as example to configure specific BGP features.

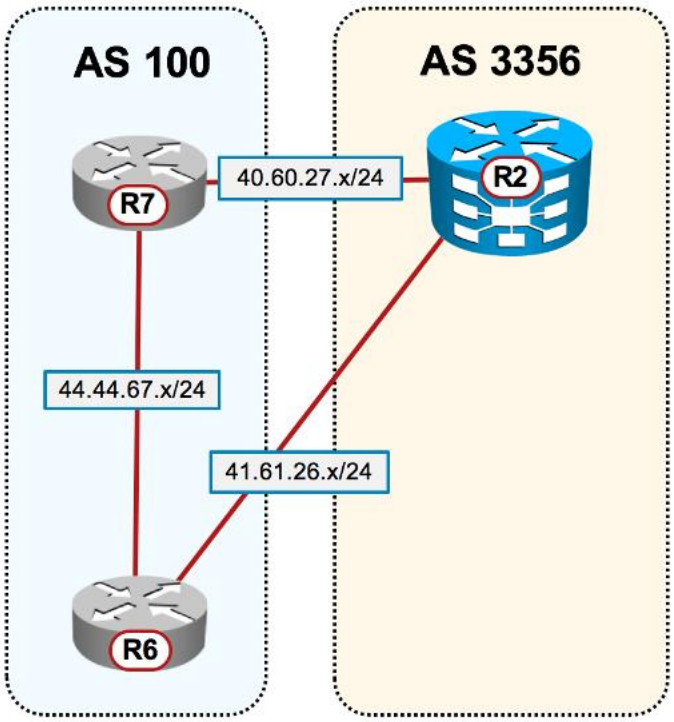
If not otherwise noted (and then – illustrated), all features work the same way for IPv4 as they do for IPv6. It will be however fair for the CCIE SP practical exam to test you on ability to configure specific features only for some specific protocol. It is therefore advised to repeat the examples using also BGP configuration for AF IPv6.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: Path selection in RR-driven BGP networks

Task 1.1: BGP multipath

BGP has capabilities to forward traffic using multiple equal and unequal cost paths. We'll use additional links between AS100 and AS3356 to demonstrate that. Let's bring up the link between CEL6 and SP1R2 and establish eBGP peering on that link using physical addresses:




```
CEL6(config)#interface gi5
CEL6(config-if)#no shutdown
CEL6(config-if)#router bgp 100
CEL6(config-router)#neighbor 41.61.26.2 remote-as 3356
CEL6(config-router)#neighbor 41.61.26.2 password LEFT
CEL6(config-router)#neighbor 2001:41:61:26::2 remote-as 3356
CEL6(config-router)#neighbor 2001:41:61:26::2 password LEFT
CEL6(config-router)#address-family ipv4 unicast
CEL6(config-router-af)#neighbor 41.61.26.2 activate
CEL6(config-router-af)#address-family ipv6 unicast
CEL6(config-router-af)#neighbor 2001:21:41:26::2 activate
CEL6(config-router-af)#^Z
```

And on SP1R2:

```
RP/0/0/CPU0:SP1R2(config)#interface gi0/0/0/3
RP/0/0/CPU0:SP1R2(config-if)#no shutdown
RP/0/0/CPU0:SP1R2(config-if)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 41.61.26.6
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#remote-as 100
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#password LEFT
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy PASS in
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy PASS out
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#nei 2001:41:61:26::6
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#remote-as 100
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#password LEFT
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy PASS in
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy PASS out
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#
```

Immediately after setting the new eBGP neighbor we can see from CEL6 side, that it has three paths to some of the prefixes: one via SP1R1, one via SP2R2 and one via it's AS100 neighbor – CEL7.

This is still however with BGP RIB – only one, best path is chosen and then candidate for router RIB.

For example, both 12.39/16 and 33.56/16 are such prefixes:

```
CEL6#sh ip bgp
[…]

     Network          Next Hop            Metric LocPrf Weight Path
 *    12.39.0.0/16     41.61.26.2                             0 3356 1239 i
 * i                   172.16.7.7               0    100      0 3356 1239 i
 *>                    41.61.16.1                             0 3356 1239 i
 *    33.56.0.0/16     41.61.26.2                             0 3356 i
 * i                   172.16.7.7               0    100      0 3356 i
 *>                    41.61.16.1                             0 3356 i
```

We can dig into some details about 12.39.0.0:

```
CEL6#sh ip bgp 12.39.0.0
BGP routing table entry for 12.39.0.0/16, version 9
Paths: (3 available, best #3, table default)
  Advertised to update-groups:
     4           5
  Refresh Epoch 1
  3356 1239, (aggregated by 1239 12.39.3.3)
    41.61.26.2 from 41.61.26.2 (33.56.2.2)
      Origin IGP, localpref 100, valid, external, atomic-aggregate
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 2
  3356 1239, (aggregated by 1239 12.39.3.3)
    172.16.7.7 (metric 2) from 172.16.7.7 (172.16.7.7)
      Origin IGP, metric 0, localpref 100, valid, internal, atomic-aggregate
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  3356 1239, (aggregated by 1239 12.39.3.3)
    41.61.16.1 from 41.61.16.1 (33.56.1.1)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, best
      rx pathid: 0, tx pathid: 0x0
```

Now, we can enable multipath for iBGP, eBGP or both iBGP and eBGP on CEL6:

```
CEL6(config)#router bgp 100
CEL6(config-router)#address-family ipv4 unicast
CEL6(config-router-af)#maximum-paths ?
  <1-32>  Number of paths
  eibgp   Both eBGP and iBGP paths as multipath
  ibgp    iBGP-multipath
CEL6(config-router-af)#maximum-paths eibgp 8
%BGP: This may cause traffic loop if not used properly (command accepted)
CEL6(config-router-af)#^Z
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński



Please note that by default if used in a form of '**maximum-paths X**', the command on IOS/IOS-XE will only enable eBGP multipathing. In such case, CEL6 would have only two multipath routes to SP1R2 as in output below:

```
CEL6#sh ip bgp
[…]
     Network          Next Hop            Metric LocPrf Weight Path
*m   12.39.0.0/16     41.61.26.2                             0 3356 1239 i
* i                   172.16.7.7               0    100      0 3356 1239 i
*>                    41.61.16.1                             0 3356 1239 i
```

You'll immediately notice, that all two other candidate paths for both 12.39/16 and 33.56/16 were marked with '**m**' – for multipath:

```
CEL6#sh ip bgp
BGP table version is 19, local router ID is 172.16.6.6
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *m   12.39.0.0/16     41.61.26.2                             0 3356 1239 i
 *mi                   172.16.7.7               0    100      0 3356 1239 i
 *>                    41.61.16.1                             0 3356 1239 i
 *m   33.56.0.0/16     41.61.26.2                             0 3356 i
 *mi                   172.16.7.7               0    100      0 3356 i
 *>                    41.61.16.1                             0 3356 i
```

And on the router route table they're all used for ECMP:

```
CEL6#sh ip route
[…]
      12.0.0.0/16 is subnetted, 1 subnets
B        12.39.0.0 [20/0] via 172.16.7.7, 00:06:22
                   [20/0] via 41.61.26.2, 00:06:22
                   [20/0] via 41.61.16.1, 00:06:22
[…]
      33.0.0.0/16 is subnetted, 1 subnets
B        33.56.0.0 [20/0] via 172.16.7.7, 00:06:22
                   [20/0] via 41.61.26.2, 00:06:22
                   [20/0] via 41.61.16.1, 00:06:22
[…]
```

Detailed view of those networks:

```
CEL6#sh ip route 12.39.0.0
Routing entry for 12.39.0.0/16
  Known via "bgp 100", distance 20, metric 0
  Tag 3356, type external
  Last update from 41.61.16.1 00:06:26 ago
  Routing Descriptor Blocks:
    172.16.7.7, from 172.16.7.7, 00:06:26 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2
      Route tag 3356
      MPLS label: none
  * 41.61.26.2, from 41.61.26.2, 00:06:26 ago
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Route metric is 0, traffic share count is 1
      AS Hops 2
      Route tag 3356
      MPLS label: none
    41.61.16.1, from 41.61.16.1, 00:06:26 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2
      Route tag 3356
      MPLS label: none
```

To get CEF forwarding information you'd do:

```
CEL6#sh ip cef 12.39.0.0 detail
12.39.0.0/16, epoch 2, flags [rib only nolabel, rib defined all labels], per-destination
sharing
  recursive via 41.61.16.1
    attached to GigabitEthernet2
  recursive via 41.61.26.2
    attached to GigabitEthernet5
  recursive via 172.16.7.7
    nexthop 44.44.67.7 GigabitEthernet3
```

Please also note that you can use '**internal**' for all low-level FIB information. In real life networks this may be used to help troubleshoot some obscure CEF misprogramming issues.

Now, as SP1R2 it's not yet multipath enabled, view from SP1R2 to a 44.44.67.0/24 prefix in AS100 shows three possible paths but only one in use – the best one:

```
RP/0/0/CPU0:SP1R2#sh bgp ipv4 unicast 44.44.67.0/24 brief
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
* i44.44.67.0/24      33.56.1.1                0    100      0 100 i
*>                    40.60.27.7               0             0 100 i
*                     41.61.26.6               0             0 100 i
```

Given this is eBGP peering, let's enable eBGP multipath on SP1R2:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-af)#maximum-paths ?
  ebgp   eBGP-multipath
  eibgp  eiBGP-multipath
  ibgp   iBGP-multipath
RP/0/0/CPU0:SP1R2(config-bgp-af)#maximum-paths ebgp 8
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
RP/0/0/CPU0:SP1R2#
```

We can see the change on SP1R2:

```
RP/0/0/CPU0:SP1R2#sh bgp ipv4 unicast 44.44.67.0/24 brief
[…]
   Network            Next Hop            Metric LocPrf Weight Path
* i44.44.67.0/24      33.56.1.1                0    100      0 100 i
*                     40.60.27.7               0             0 100 i
*>                    41.61.26.6               0             0 100 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Note that IOS XR doesn't show multipath entries in the output, but routing table and CEF confirm our configuration, with both external BGP peers used as multipath:

```
RP/0/0/CPU0:SP1R2#sh route ipv4 unicast 44.44.67.0/24
Routing entry for 44.44.67.0/24
  Known via "bgp 3356", distance 20, metric 0
  Tag 100, type external
  Installed Jan  1 12:06:11.793 for 00:01:01
  Routing Descriptor Blocks
    40.60.27.7, from 40.60.27.7, BGP external, BGP multi path
      Route metric is 0
    41.61.26.6, from 41.61.26.6, BGP external, BGP multi path
      Route metric is 0
  No advertising protos.
RP/0/0/CPU0:SP1R2#sh cef ipv4 44.44.67.0/24
44.44.67.0/24, version 96, internal 0x5000001 0x0 (ptr 0xa1408df4) [1], 0x0 (0x0), 0x0
(0x0)
 Updated Jan  1 12:06:11.813
 local adjacency 40.60.27.7
 Prefix Len 24, traffic index 0, precedence n/a, priority 4
   via 40.60.27.7/32, 2 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]
    path-idx 0 NHID 0x0 [0xa1408374 0x0]
    next hop 40.60.27.7/32 via 40.60.27.7/32
   via 41.61.26.6/32, 2 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]
    path-idx 1 NHID 0x0 [0xa14093f4 0x0]
    next hop 41.61.26.6/32 via 41.61.26.6/32
```



In this scenario we're dealing with the same neighbor AS and the same AS_PATH length. To satisfy multipath requirements, prefix needs to have by default:

- the same weight
- the same local preference
- the same AS_PATH (entire attribute, not just length)
- the same origin code
- the same MED
- the same IGP distance
- different nexthop

For situations where the AS would be different (or AS_PATH would have different length) we have configuration knobs that can be applied to still make multipath work.

In IOS and IOS-XE those commands are hidden and appear in the config only when you type them explicitly. In IOS XR they are visible in the parser and in the helper as well.

This form will ignore AS_PATH length:

```
bgp bestpath as-path ignore
```

This form (they're exclusive) will keep the requirement to keep the AS_PATH length the same, but it may have different contents:

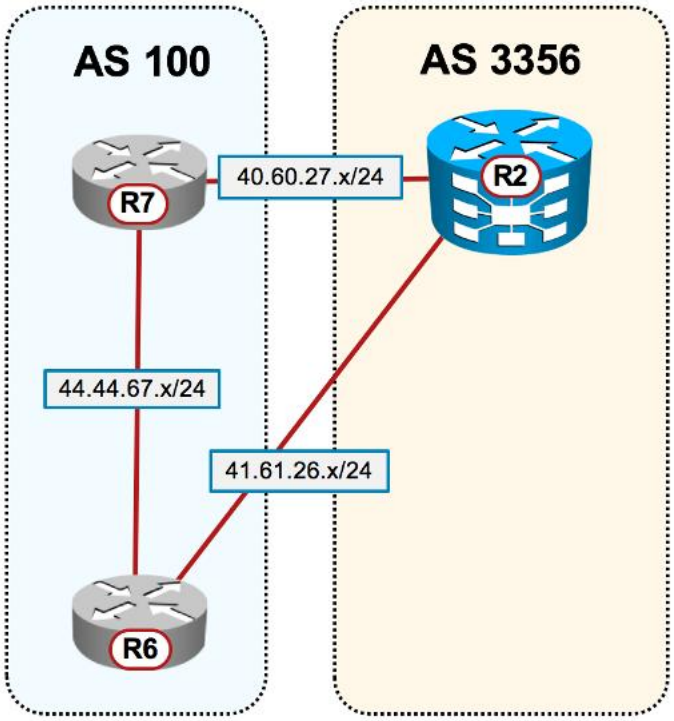
```
bgp bestpath as-path multipath-relax
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: BGP DMZ link bandwidth on IOS-XE

In situations where ECMP is not possible or even harmful as underlying infrastructure doesn't offer the same bandwidth on interfaces, you can use BGP DMZ link bandwidth to execute UCMP (Unequal Cost Multi Path).

We'll use the configuration made in previous task and the same fragment of bigger topology to showcase the feature:



On R6 and R7 enable BGP DMZ link bandwidth and enable sending extended communities towards each other router in AS100.

First, let's enable DMZ link bandwidth on CEL7:

```
CEL7#conf t
CEL7(config)#router bgp 100
CEL7(config-router)#bgp dmzlink-bw
CEL7(config-router)#address-family ipv4 unicast
CEL7(config-router-af)#neighbor 40.60.27.2 dmzlink-bw
CEL7(config-router-af)#exit
CEL7(config-router)#address-family ipv6 unicast
CEL7(config-router-af)#neighbor 2001:40:60:27::2 dmzlink-bw
CEL7(config-router-af)#^Z
```

Let's verify if the bandwidth is properly adjusted:

```
CEL7#sh bgp 12.39.0.0
BGP routing table entry for 12.39.0.0/16, version 29
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  3356 1239, (aggregated by 1239 12.39.3.3)
    172.16.6.6 (metric 301) from 172.16.6.6 (172.16.6.6)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Origin IGP, metric 0, localpref 100, valid, internal, atomic-aggregate
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  3356 1239, (aggregated by 1239 12.39.3.3)
    40.60.27.2 from 40.60.27.2 (33.56.2.2)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, best
      DMZ-Link Bw 125000 kbytes
      rx pathid: 0, tx pathid: 0x0
```

As you can see, path from CEL6 is not marked – and that's because CEL6 is not yet activated for DMZ link bandwidth feature, and we're not passing extended communities between IOS XE routers to signal this attribute. Let's fix that – on CEL6:

```
CEL6#conf t
CEL6(config)#router bgp 100
CEL6(config-router)#bgp dmzlink-bw
CEL6(config-router)#address-family ipv4 unicast
CEL6(config-router-af)#neighbor 41.61.16.1 dmzlink-bw
CEL6(config-router-af)#neighbor 41.61.26.2 dmzlink-bw
CEL6(config-router-af)#exit
CEL6(config-router)#address-family ipv6 unicast
CEL6(config-router-af)#neighbor 2001:41:61:16::1 dmzlink-bw
CEL6(config-router-af)#neighbor 2001:41:61:26::2 dmzlink-bw
CEL6(config-router-af)#^Z
```

Note we have not yet configured extended communities between CEL6 and CEL7. From CEL6 perspective we should have two links (as we have two eBGP sessions) marked with DMZ link bandwidth:

```
CEL6#sh bgp 33.56.0.0
BGP routing table entry for 33.56.0.0/16, version 48
Paths: (3 available, best #3, table default)
Multipath: eiBGP
  Advertised to update-groups:
     1           2
  Refresh Epoch 1
  3356, (aggregated by 3356 33.56.2.2)
    41.61.26.2 from 41.61.26.2 (33.56.2.2)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, multipath
      DMZ-Link Bw 125000 kbytes
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  3356, (aggregated by 3356 33.56.2.2)
    172.16.7.7 (metric 2) from 172.16.7.7 (172.16.7.7)
      Origin IGP, metric 0, localpref 100, valid, internal, atomic-aggregate,
multipath(oldest)
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  3356, (aggregated by 3356 33.56.1.1)
    41.61.16.1 from 41.61.16.1 (33.56.1.1)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, multipath, best
      DMZ-Link Bw 125000 kbytes
      rx pathid: 0, tx pathid: 0x0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

As the bandwidth inherited from the default configuration of GigabitEthernet interfaces is the same for both links, we should still be load balancing equally between both external and one internal link. Let's check that:

```
CEL6#sh ip route 33.56.0.0
Routing entry for 33.56.0.0/16
  Known via "bgp 100", distance 20, metric 0
  Tag 3356, type external
  Last update from 41.61.16.1 00:13:31 ago
  Routing Descriptor Blocks:
    172.16.7.7, from 172.16.7.7, 00:13:31 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 3356
      MPLS label: none
  * 41.61.26.2, from 41.61.26.2, 00:13:31 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 3356
      MPLS label: none
    41.61.16.1, from 41.61.16.1, 00:13:31 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 3356
      MPLS label: none
```

Now, we'll adjust link between CEL6 and SP1R1 (so Gi2):

```
CEL6#conf t
CEL6(config)#int gi2
CEL6(config-if)#bandwidth 100000
CEL6(config-if)#^Z
```

The DMZ link bandwidth was immediately adjusted:

```
CEL6#sh ip bgp 33.56.0.0
BGP routing table entry for 33.56.0.0/16, version 60
Paths: (3 available, best #3, table default)
Multipath: eiBGP
  Advertised to update-groups:
     1           2
  Refresh Epoch 1
  3356, (aggregated by 3356 33.56.2.2)
    41.61.26.2 from 41.61.26.2 (33.56.2.2)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, multipath
      DMZ-Link Bw 125000 kbytes ! it's 1Gbps
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  3356, (aggregated by 3356 33.56.2.2)
    172.16.7.7 (metric 2) from 172.16.7.7 (172.16.7.7)
      Origin IGP, metric 0, localpref 100, valid, internal, atomic-aggregate,
multipath(oldest)
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  3356, (aggregated by 3356 33.56.1.1)
    41.61.16.1 from 41.61.16.1 (33.56.1.1)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, multipath, best
      DMZ-Link Bw 12500 kbytes ! it's 100Mbps as adjusted above
      rx pathid: 0, tx pathid: 0x0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Which means traffic share will also change:

```
CEL6#sh ip route 33.56.0.0
Routing entry for 33.56.0.0/16
  Known via "bgp 100", distance 20, metric 0
  Tag 3356, type external
  Last update from 41.61.16.1 00:01:53 ago
  Routing Descriptor Blocks:
    172.16.7.7, from 172.16.7.7, 00:01:53 ago
      Route metric is 0, traffic share count is 10
      AS Hops 1
      Route tag 3356
      MPLS label: none
  * 41.61.26.2, from 41.61.26.2, 00:01:53 ago
      Route metric is 0, traffic share count is 10
      AS Hops 1
      Route tag 3356
      MPLS label: none
    41.61.16.1, from 41.61.16.1, 00:01:53 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 3356
      MPLS label: none
```

Note that 41.61.16.1 was downgraded from 1000Mbps to 100Mbps, and that means effectively it will share 1/10th of the other links.

If you want to take a look how CEF will do the data-plane load balancing, it's worth to check the output of:

```
CEL6#sh ip cef 33.56.0.0 internal | b output chain
  output chain:
    loadinfo 80007F49BA169A18, per-session, 3 choices, flags 0103, 13 locks
      flags [Per-session, for-rx-IPv4, indirection]
      12 hash buckets
        < 0 > IP adj out of GigabitEthernet2, addr 41.61.16.1 7F49C05C22B0
        < 1 > IP adj out of GigabitEthernet5, addr 41.61.26.2 7F49C05C1CE0
        < 2 > IP adj out of GigabitEthernet3, addr 44.44.67.7 7F49C05C2880
        < 3 > IP adj out of GigabitEthernet5, addr 41.61.26.2 7F49C05C1CE0
        < 4 > IP adj out of GigabitEthernet5, addr 41.61.26.2 7F49C05C1CE0
        < 5 > IP adj out of GigabitEthernet5, addr 41.61.26.2 7F49C05C1CE0
        < 6 > IP adj out of GigabitEthernet5, addr 41.61.26.2 7F49C05C1CE0
        < 7 > IP adj out of GigabitEthernet5, addr 41.61.26.2 7F49C05C1CE0
        < 8 > IP adj out of GigabitEthernet5, addr 41.61.26.2 7F49C05C1CE0
        < 9 > IP adj out of GigabitEthernet5, addr 41.61.26.2 7F49C05C1CE0
        <10 > IP adj out of GigabitEthernet5, addr 41.61.26.2 7F49C05C1CE0
        <11 > IP adj out of GigabitEthernet5, addr 41.61.26.2 7F49C05C1CE0
      Subblocks:
        None
```

So while the 'ip route' traffic share count shows 10 even for the internal link to CEL7 (and we haven't yet enabled exchange of DMZ link bandwidth using extended communities), the CEF gives more detailed view: 10 entries pointing to next-hop of 41.61.26.2, 1 entry pointing to 41.61.16.1 and one entry pointing to 44.44.67.7 (iBGP session).

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Now, let's enable exchange of extended communities between CEL6 and CEL7:

```
CEL6(config)#router bgp 100
CEL6(config-router)#address-family ipv4
CEL6(config-router-af)#neighbor 172.16.7.7 send-community both
CEL6(config-router-af)#address-family ipv6
CEL6(config-router-af)#neighbor 2001:44:66:CC::7 send-community both
CEL7(config)#router bgp 100
CEL7(config-router)#address-family ipv4
CEL7(config-router-af)#neighbor 172.16.6.6 send-community both
CEL7(config-router-af)#address-family ipv6
CEL7(config-router-af)#neighbor 2001:44:66:CC::6 send-community both
```

Even the internally exchanged route has properly attached DMZ link bandwidth extended community:

```
CEL6#sh ip bgp 33.56.0.0
BGP routing table entry for 33.56.0.0/16, version 64
Paths: (3 available, best #3, table default)
Multipath: eiBGP
  Advertised to update-groups:
     2           3
  Refresh Epoch 1
  3356, (aggregated by 3356 33.56.2.2)
    41.61.26.2 from 41.61.26.2 (33.56.2.2)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, multipath
      DMZ-Link Bw 125000 kbytes
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  3356, (aggregated by 3356 33.56.2.2)
    172.16.7.7 (metric 2) from 172.16.7.7 (172.16.7.7)
      Origin IGP, metric 0, localpref 100, valid, internal, atomic-aggregate,
multipath(oldest)
      DMZ-Link Bw 125000 kbytes
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  3356, (aggregated by 3356 33.56.1.1)
    41.61.16.1 from 41.61.16.1 (33.56.1.1)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, multipath, best
      DMZ-Link Bw 12500 kbytes
      rx pathid: 0, tx pathid: 0x0
```

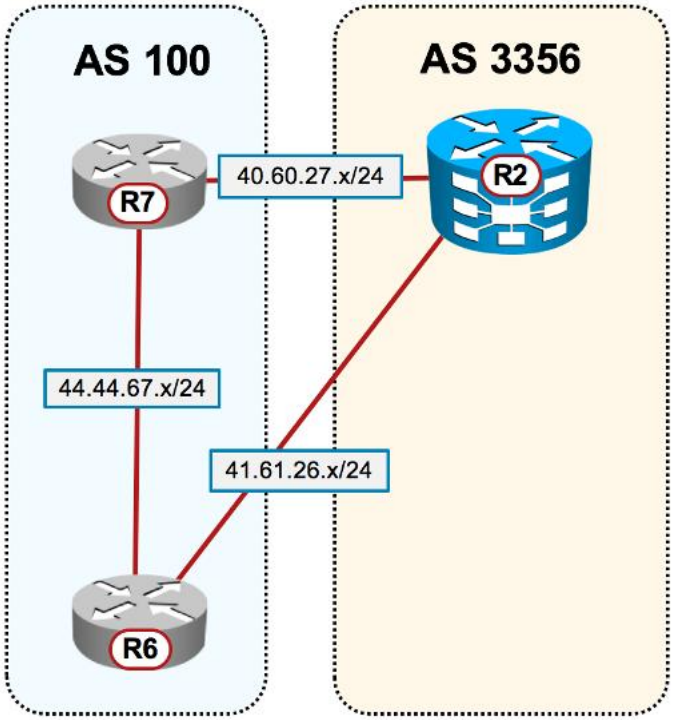
This concludes task as it relates to DMZ link bandwidth on IOS-XE.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3: BGP DMZ link bandwidth on IOS XR

We'll execute similar configuration for IOS XR. There's a difference in the configuration, as for IOS XR you need to use RPL policies for incoming eBGP updates to properly program dynamic bandwidth.

Let us start right where we left off during Task 1.2:



IPv4 and IPv6 AFs on SP1R2 should be already enabled for eiBGP multipathing. If that's not true, please make sure starting configuration reflects this config snippet:

```
RP/0/0/CPU0:SP1R2#sh running-config router bgp
router bgp 3356
 bgp router-id 33.56.2.2
 address-family ipv4 unicast
  maximum-paths ebgp 8 ! note that as of 6.1.3 and earlier releases, DMZ link
                       ! won't work correctly if done for eiBGP multipathing
  aggregate-address 33.56.0.0/16 summary-only
  redistribute connected route-policy RED-BGP
 !
 address-family ipv6 unicast
  maximum-paths ebgp 8
  aggregate-address 2001:33:56::/48 summary-only
  redistribute connected route-policy RED-BGP
 !
```

We'll want to load balance using ratio of 9:1 for a link towards R7 and R6 respectively. So, we need to configure two separate inbound policies:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#route-policy DMZ-BAND-R7
RP/0/0/CPU0:SP1R2(config-rpl)#set extcommunity bandwidth ( 3356:9000 )
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
RP/0/0/CPU0:SP1R2(config)#route-policy DMZ-BAND-R6
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R2(config-rpl)#set extcommunity bandwidth ( 3356:1000 )
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
RP/0/0/CPU0:SP1R2(config)#commit
```

Those policies needs to be attached to neighbors, along with activation of the DMZ link bandwidth feature:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#dmz-link-bandwidth
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy DMZ-BAND-R7 in
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#send-community-ebgp
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#send-extended-community-ebgp
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#exit
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#exit
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 41.61.26.6
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#dmz-link-bandwidth
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy DMZ-BAND-R6 in
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#send-community-ebgp
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#send-extended-community-ebgp
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#commit
```



As a reminder:

IOS and IOS-XE by default doesn't send any communities. Enabling:

```
neighbor X.X.X.X send-community ! for standard communities
neighbor X.X.X.X send-community extended ! for extended
communities
```

…or if you want to send both:

```
neighbor X.X.X.X send-community both ! for both types
```

IOS XR sends both standard and extended communities on iBGP sessions, while on eBGP this requires explicit configuration (as above):

```
send-community-ebgp ! for standard communities
send-extended-community-ebgp ! for extended communities
```

You can see that the extended community was attached for prefixes advertised by AS100. The weighting is automatically recalculated by IOS XR to 72:8:

```
RP/0/0/CPU0:SP1R2#sh bgp 44.44.67.0
  Path #1: Received by speaker 0
  Not advertised to any peer
  100
    33.56.1.1 (metric 2) from 33.56.5.5 (33.56.1.1)
[…]
  Path #2: Received by speaker 0
[…]
    40.60.27.7 from 40.60.27.7 (172.16.7.7)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Origin IGP, metric 0, localpref 100, valid, external, multipath
      Received Path ID 0, Local Path ID 0, version 0
      Extended community: LB:3356:72
      Origin-AS validity: not-found
  Path #3: Received by speaker 0
[…]
  100
    41.61.26.6 from 41.61.26.6 (172.16.6.6)
      Origin IGP, metric 0, localpref 100, valid, external, best, group-best, multipath
      Received Path ID 0, Local Path ID 0, version 95
      Extended community: LB:3356:8
      Origin-AS validity: not-found
```

We can also check how forwarding will split the traffic load:

```
RP/0/0/CPU0:SP1R2#sh route 44.44.67.0/24

Routing entry for 44.44.67.0/24
  Known via "bgp 3356", distance 20, metric 0
  Tag 100, type external
  Installed Jan  2 00:42:28.188 for 00:00:13
  Routing Descriptor Blocks
    40.60.27.7, from 40.60.27.7, BGP external, BGP multi path
      Route metric is 0, Wt is 72
    41.61.26.6, from 41.61.26.6, BGP external, BGP multi path
      Route metric is 0, Wt is 8
  No advertising protos.
```

And how CEF will split the buckets:

```
RP/0/0/CPU0:SP1R2#sh cef 44.44.67.0/24 detail
44.44.67.0/24, version 246, internal 0x1000001 0x0 (ptr 0xa1400ff4) [1], 0x0 (0x0), 0x0
(0x0)
[…]
    Weight distribution:
    slot 0, weight 72, normalized_weight 9, class 0
    slot 1, weight 8, normalized_weight 1, class 0

    Load distribution: 0 0 0 0 0 0 0 0 0 1 (refcount 1)

    Hash  OK  Interface                 Address
    0     Y   GigabitEthernet0/0/0/1    40.60.27.7
    1     Y   GigabitEthernet0/0/0/1    40.60.27.7
    2     Y   GigabitEthernet0/0/0/1    40.60.27.7
    3     Y   GigabitEthernet0/0/0/1    40.60.27.7
    4     Y   GigabitEthernet0/0/0/1    40.60.27.7
    5     Y   GigabitEthernet0/0/0/1    40.60.27.7
    6     Y   GigabitEthernet0/0/0/1    40.60.27.7
    7     Y   GigabitEthernet0/0/0/1    40.60.27.7
    8     Y   GigabitEthernet0/0/0/1    40.60.27.7 ! note this is 9:1, exactly as
    9     Y   GigabitEthernet0/0/0/3    41.61.26.6 ! configured
```

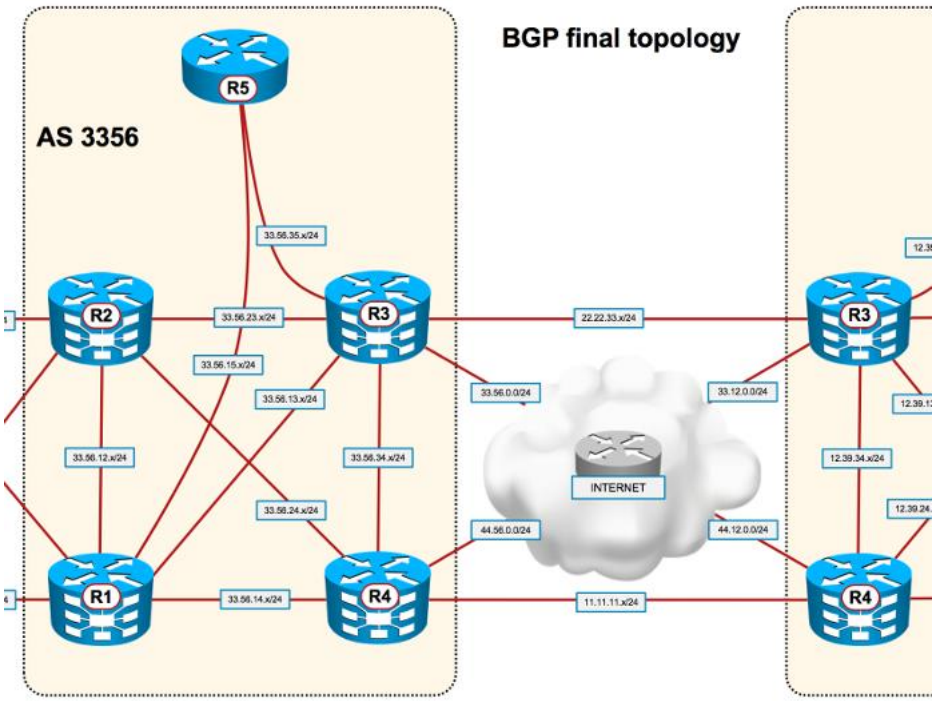
This concludes DMZ link bandwidth configuration. Please roll back any changes from Task 1.3 and Task 1.2.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4: BGP AddPath extension

As BGP selects only best path to advertise to other peers, let's use Add-Path extensions to overcome this limitation in our Route Reflector design.

Our goal will be to enable R1 to load-share traffic to 12.39/16:



First, the RR hides redundant paths to routers, as it runs its own best-path algorithm and reflects the one it choose to all its RR-clients.

For example, observe that from SP1R1 perspective, 12.39/16 prefix (AS1239) is visible only via 33.56.3.3 (SP1R3 Loopback) and the second potential path (via SP1R4) is not:

```
RP/0/0/CPU0:SP1R1#sh ip bgp 12.39.0.0
BGP routing table entry for 12.39.0.0/16
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                319         319
Last Modified: Jan  1 19:13:20.702 for 00:36:29
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    41.61.16.6
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    41.61.16.6
  1239, (aggregated by 1239 12.39.3.3)
    33.56.3.3 (metric 2) from 33.56.5.5 (33.56.3.3)
      Origin IGP, localpref 100, valid, internal, atomic-aggregate, best, group-best
      Received Path ID 0, Local Path ID 0, version 319
      Originator: 33.56.3.3, Cluster list: 33.56.5.5
```

The RR (SP1R5) sees all the paths:

```
SP1-R5-RR#sh ip bgp 12.39.0.0
BGP routing table entry for 12.39.0.0/16, version 2
Paths: (2 available, best #2, table default)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Multipath: iBGP
  Advertised to update-groups:
     11
  Refresh Epoch 1
  1239, (aggregated by 1239 12.39.4.4), (Received from a RR-client)
    33.56.4.4 (metric 3) from 33.56.4.4 (33.56.4.4)
      Origin IGP, localpref 100, valid, internal, atomic-aggregate
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  1239, (aggregated by 1239 12.39.3.3), (Received from a RR-client)
    33.56.3.3 (metric 2) from 33.56.3.3 (33.56.3.3)
      Origin IGP, localpref 100, valid, internal, atomic-aggregate, best
      rx pathid: 0, tx pathid: 0x0
```

The “best” marked path is selected as one and then reflected to all clients.

```
RP/0/0/CPU0:SP1R2#sh ip route | i 12.39.0
B    12.39.0.0/16 [200/0] via 33.56.3.3, 00:39:01
```

To add AddPaths, we need to enable RR to receive and send all additional paths:

```
SP1-R5-RR(config)#router bgp 3356
SP1-R5-RR(config-router)#address-family ipv4 unicast
SP1-R5-RR(config-router-af)#bgp additional-paths send receive
SP1-R5-RR(config-router-af)#bgp additional-paths select all
SP1-R5-RR(config-router-af)#neighbor 33.56.1.1 additional-paths send receive
SP1-R5-RR(config-router-af)#neighbor 33.56.1.1 advertise additional-paths all
SP1-R5-RR(config-router-af)#neighbor 33.56.2.2 additional-paths send receive
SP1-R5-RR(config-router-af)#neighbor 33.56.2.2 advertise additional-paths all
SP1-R5-RR(config-router-af)#neighbor 33.56.3.3 additional-paths send receive
SP1-R5-RR(config-router-af)#neighbor 33.56.3.3 advertise additional-paths all
SP1-R5-RR(config-router-af)#neighbor 33.56.4.4 additional-paths send receive
SP1-R5-RR(config-router-af)#neighbor 33.56.4.4 advertise additional-paths all
SP1-R5-RR(config-router)#address-family ipv6 unicast
SP1-R5-RR(config-router-af)#bgp additional-paths send receive
SP1-R5-RR(config-router-af)#bgp additional-paths select all
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:AA::1 additional-paths send receive
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:AA::1 advertise additional-paths all
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:AA::2 additional-paths send receive
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:AA::2 advertise additional-paths all
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:AA::3 additional-paths send receive
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:AA::3 advertise additional-paths all
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:AA::4 additional-paths send receive
SP1-R5-RR(config-router-af)#neighbor 2001:33:56:AA::4 advertise additional-paths all
```

Notice, that during the changes, BGP sessions will flap. This is needed to negotiate the additional capabilities on the session.

After adding the AddPath configuration and setting the policy to select ‘**all**’ possible paths, the routing entry for 12.39/16 from BGP perspective will change:

```
SP1-R5-RR#sh ip bgp 12.39.0.0
BGP routing table entry for 12.39.0.0/16, version 46
Paths: (2 available, best #2, table default)
  Path not advertised to any peer
  Refresh Epoch 1
  1239, (aggregated by 1239 12.39.4.4), (Received from a RR-client)
    33.56.4.4 (metric 3) from 33.56.4.4 (33.56.4.4)
      Origin IGP, localpref 100, valid, internal, atomic-aggregate, all
      rx pathid: 0, tx pathid: 0x1
  Path advertised to update-groups:
     2
  Refresh Epoch 1
  1239, (aggregated by 1239 12.39.3.3), (Received from a RR-client)
    33.56.3.3 (metric 2) from 33.56.3.3 (33.56.3.3)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Origin IGP, localpref 100, valid, internal, atomic-aggregate, best
      rx pathid: 0, tx pathid: 0x0
```

Notice that 12.39/16 is now listed twice – as bestpath and as add-path path:

```
SP1-R5-RR#sh ip bgp
     Network          Next Hop            Metric LocPrf Weight Path
 * ia 12.39.0.0/16     33.56.4.4                    100      0 1239 i
 *>i                   33.56.3.3                    100      0 1239 i
 […]
```

However from SP1R1 perspective without additional configuration, we're still receiving only one path as we're not signalling capability to work with AddPath extension, we're only receiving that from SP1R5:

```
RP/0/0/CPU0:SP1R1#sh bgp ipv4 unicast neighbors 33.56.5.5
BGP neighbor is 33.56.5.5
 Remote AS 3356, local AS 3356, internal link
 Remote router ID 33.56.5.5
  BGP state = Established, up for 00:01:28
[…]
For Address Family: IPv4 Unicast
[…]
  AF-dependent capabilities:
    Additional-paths Send: received
    Additional-paths Receive: received
  Route refresh request: received 0, sent 0
```

So, we'll modify capabilities of receive and then install (select) BGP paths from SP1R5.

```
RP/0/0/CPU0:SP1R1(config)#route-policy BGP-ADD-PATH
RP/0/0/CPU0:SP1R1(config-rpl)#set path-selection all advertise
RP/0/0/CPU0:SP1R1(config-rpl)#exit
RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-af)#additional-paths selection route-policy BGP-ADD-PATH
RP/0/0/CPU0:SP1R1(config-bgp-af)#additional-paths receive
RP/0/0/CPU0:SP1R1(config-bgp-af)#maximum-paths ibgp 8
```

Note, that from SP1R1 perspective those paths had the same metric (to BGP next-hop, which is SP1R3 and SP1R4 loopback addresses), so we don't need to play with '**maximum-paths ibgp 8 unequal-cost**'). Let's check the configuration effect:

```
RP/0/0/CPU0:SP1R1#sh ip bgp 12.39.0.0
BGP routing table entry for 12.39.0.0/16
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker               558         558
Last Modified: Jan  1 20:50:09.702 for 00:00:33
Paths: (2 available, best #1)
  Advertised to peers (in unique update groups):
    41.61.16.6
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    41.61.16.6
  1239, (aggregated by 1239 12.39.3.3)
    33.56.3.3 (metric 2) from 33.56.5.5 (33.56.3.3)
      Origin IGP, localpref 100, valid, internal, atomic-aggregate, best, group-best,
multipath
      Received Path ID 0, Local Path ID 0, version 558
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

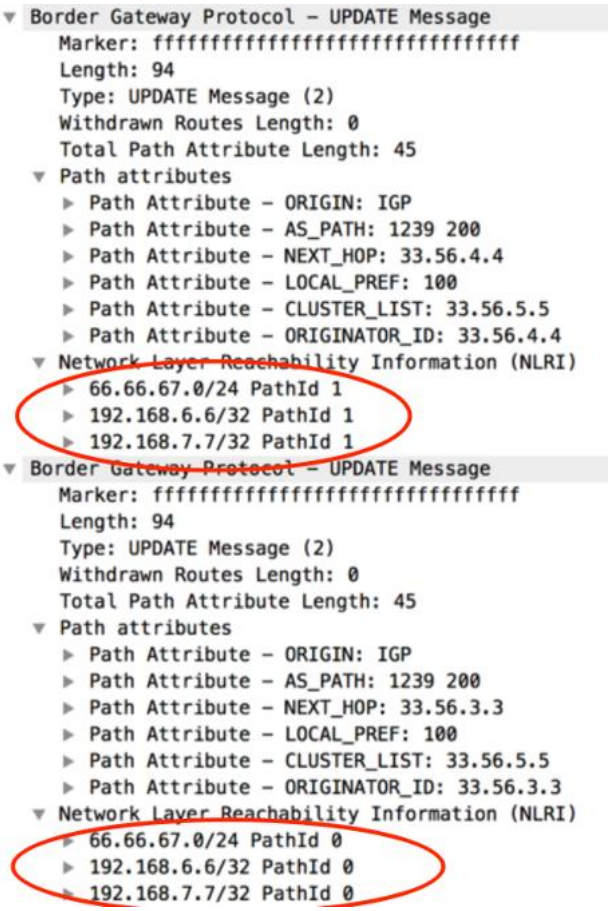
```
      Originator: 33.56.3.3, Cluster list: 33.56.5.5
  Path #2: Received by speaker 0
  Not advertised to any peer
  1239, (aggregated by 1239 12.39.4.4)
    33.56.4.4 (metric 2) from 33.56.5.5 (33.56.4.4)
      Origin IGP, localpref 100, valid, internal, atomic-aggregate, multipath, add-path
      Received Path ID 1, Local Path ID 2, version 558
      Originator: 33.56.4.4, Cluster list: 33.56.5.5
```

We can also verify, that traffic to 12.39.0.0/16 will be CEF load-balanced:

```
RP/0/0/CPU0:SP1R1#sh cef ipv4 12.39.0.0
12.39.0.0/16, version 769, internal 0x5000001 0x0 (ptr 0xa1408974) [1], 0x0 (0x0), 0x0
(0x0)

 local adjacency 33.56.13.3
 Prefix Len 16, traffic index 0, precedence n/a, priority 4
   via 33.56.3.3/32, 2 dependencies, recursive, bgp-multipath [flags 0x6080]
    path-idx 0 NHID 0x0 [0xa1408c74 0x0]
    next hop 33.56.3.3/32 via 33.56.3.3/32
   via 33.56.4.4/32, 2 dependencies, recursive, bgp-multipath [flags 0x6080]
    path-idx 1 NHID 0x0 [0xa1408ef4 0x0]
    next hop 33.56.4.4/32 via 33.56.4.4/32
```

You can see also in packet capture, we're getting unique PathIDs for the same prefixes with different next-hops from RR (33.56.5.5) - via 33.56.4.4 (R4) and 33.56.3.3 (R3).



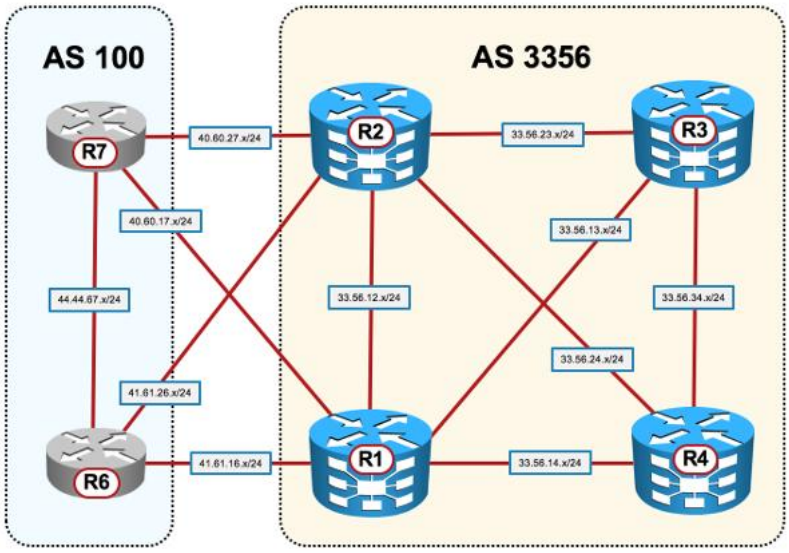
Please remove any AddPath configuration before proceeding to the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.5: BGP ORR: Optimal Route Reflection

ORR brings capability to use existing IGP information into BGP Route Reflector best path decision process.

Based on defined groups (that are made of routers for which specific view should be configured) BGP RR will calculate best paths from point of view of group, not from point of view of RR itself.



For this task we will reconfigure AS3356 with SP1R4 acting as route-reflector. You can use IOS XR commit/rollback capability to get back with old configuration once we finish the task.

For SP1R5 we'll just shut down its Gi2 and Gi3 interfaces, to close the iBGP sessions – that's the fastest way to get it out of the picture and at the same time be able to get back to it once we finish:

```
SP1-R5-RR(config)#int gi2
SP1-R5-RR(config-if)#shut
SP1-R5-RR(config-if)#int gi3
SP1-R5-RR(config-if)#shut
SP1-R5-RR(config-if)#^Z
```

Now, on each of the other routers in AS3356, we'll shut down the neighbors leading to AS3356 and instead configure iBGP session to SP1R4. For example, on SP1R1:

```
RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#neighbor 33.56.5.5
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#shutdown
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#nei 2001:33:56:aa::5
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#shut
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#exit
RP/0/0/CPU0:SP1R1(config-bgp)#exit
RP/0/0/CPU0:SP1R1(config)#commit label ORR
```

Notice that our config was committed with easily recognizable label, that can be referenced later on to get back to "clean" config.

```
RP/0/0/CPU0:SP1R4(config)#router bgp 3356
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R4(config-bgp)#neighbor 33.56.5.5
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#shut
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#nei 2001:33:56:aa::5
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#shut
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#commit label ORR
RP/0/0/CPU0:SP1R4(config-bgp)#neighbor-group AS3356
RP/0/0/CPU0:SP1R4(config-bgp-nbrgrp)#update-source loopback 0
RP/0/0/CPU0:SP1R4(config-bgp-nbrgrp)#remote-as 3356
RP/0/0/CPU0:SP1R4(config-bgp-nbrgrp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R4(config-bgp-nbrgrp-af)#route-reflector-client
RP/0/0/CPU0:SP1R4(config-bgp-nbrgrp-af)#exit
RP/0/0/CPU0:SP1R4(config)#router bgp 3356
RP/0/0/CPU0:SP1R4(config-bgp)#neighbor 33.56.1.1
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#use neighbor-group AS3356
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#neighbor 33.56.2.2
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#use neighbor-group AS3356
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#neighbor 33.56.3.3
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#use neighbor-group AS3356
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

The problem with this setup (as usual with RRs) is that every path sent from RR is the best one from RR point of view. For example, let's take a look at what information SP1R4 acting as RR is providing to SP1R3 to get to the AS100 loopback prefixes:

```
RP/0/0/CPU0:SP1R3#sh ip bgp
[…]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
[…]
*>i172.16.6.6/32      33.56.1.1                0    100      0 100 i
*>i172.16.7.7/32      33.56.1.1                     100      0 100 i
[…]
```

CEF confirms, that even though R3 can easily reach the loopbacks via R2 and that would be shorter, it'll use R1 as exit point from AS3356:

```
RP/0/0/CPU0:SP1R3#sh ip cef 172.16.6.6
172.16.6.6/32, version 179, internal 0x5000001 0x0 (ptr 0xa14235f4) [1], 0x0 (0x0), 0x0
(0x0)
 Updated Jan  1 16:43:08.055
 local adjacency 33.56.13.1
 Prefix Len 32, traffic index 0, precedence n/a, priority 4
   via 33.56.1.1/32, 2 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa1422874 0x0]
    next hop 33.56.1.1/32 via 33.56.1.1/32
RP/0/0/CPU0:SP1R3#sh ip cef 172.16.7.7
172.16.7.7/32, version 181, internal 0x5000001 0x0 (ptr 0xa1423674) [1], 0x0 (0x0), 0x0
(0x0)
 Updated Jan  1 16:43:08.056
 local adjacency 33.56.13.1
 Prefix Len 32, traffic index 0, precedence n/a, priority 4
   via 33.56.1.1/32, 2 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa1422874 0x0]
    next hop 33.56.1.1/32 via 33.56.1.1/32
```

So, for the traffic from SP1R3 loopback to CEL7, the actual path will be SP1R1, CEL6 and only then CEL7 instead of closer SP1R2 — CEL7:

```
RP/0/0/CPU0:SP1R3#traceroute 172.16.7.7 source 33.56.3.3
[…]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Type escape sequence to abort.
Tracing the route to 172.16.7.7

 1   33.56.13.1 9 msec   0 msec   0 msec ! inbound interface on SP1R1
 2   41.61.16.6 0 msec   9 msec   0 msec ! inbound interface on CEL6
 3   44.44.67.7 9 msec   *   0 msec      ! the target – interface of CEL7
```

Even if we take the SP1R1 — SP1R3 link out of work:

```
RP/0/0/CPU0:SP1R3(config)#int gi0/0/0/6
RP/0/0/CPU0:SP1R3(config-if)#shut
RP/0/0/CPU0:SP1R3(config-if)#
```

…from SP1R4 still the best path to AS100 networks is via SP1R1 not via SP1R2.

So - let's configure a group for SP1R3 and ask SP1R4 (RR) to compute the best paths from SP1R3 point of view and then send them to it:

```
RP/0/0/CPU0:SP1R4(config)#router bgp 3356
RP/0/0/CPU0:SP1R4(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R4(config-bgp-af)#optimal-route-reflection SP1R3 33.56.3.3
RP/0/0/CPU0:SP1R4(config-bgp-af)#exit
RP/0/0/CPU0:SP1R4(config-bgp)#neighbor 33.56.3.3
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R4(config-bgp-nbr-af)#optimal-route-reflection SP1R3
RP/0/0/CPU0:SP1R4(config-bgp-nbr-af)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

SP1R3 is now configured to serve SP1R3 group to neighbor 33.56.3.3:

```
RP/0/0/CPU0:SP1R4#sh ip bgp nei 33.56.3.3
[…]
BGP neighbor is 33.56.3.3
 Remote AS 3356, local AS 3356, internal link
 Remote router ID 33.56.3.3
 Cluster ID 33.56.4.4
  BGP state = Established, up for 00:01:53
[…]
 For Address Family: IPv4 Unicast
  BGP neighbor version 126
  Update group: 0.2 Filter-group: 0.4  No Refresh request being processed
  Route-Reflector Client
  ORR root (configured): SP1R3; Index: 0
```

Because we're using OSPFv2/v3 as underlying IGP, we need to also configure the IGP to distribute BGP Link State information, needed by ORR to calculate the routes (ORR uses Type 10 LSAs). For that, we also need to enable MPLS Traffic Engineering feature – what's needed is just to assign the interfaces to TE capabilities, not enable whole MPLS stack itself. So for each node we need to:

1. Tell OSPF to carry LS extensions (in opaque LSA 10)
2. Tell OSPF to use MPLS TE extensions
3. Assign the MPLS TE feature to interfaces, so OSPF can read the interface data

For SP1R1 this would need:

```
RP/0/0/CPU0:SP1R4(config)#router ospf AS3356
RP/0/0/CPU0:SP1R4(config-ospf)#distribute bgp-ls
RP/0/0/CPU0:SP1R4(config-ospf)#area 0
RP/0/0/CPU0:SP1R4(config-ospf-ar)#mpls traffic-eng
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
! now enable the MPLS-TE without actually configuring whole MPLS stack
! MPLS-TE is needed for OSPF/ISIS to signal link parameters up to BGP

RP/0/0/CPU0:SP1R1(config)#mpls traffic-eng
RP/0/0/CPU0:SP1R1(config-mpls-te)#interface gigabitEthernet 0/0/0/0
RP/0/0/CPU0:SP1R1(config-mpls-te)#interface gigabitEthernet 0/0/0/2
RP/0/0/CPU0:SP1R1(config-mpls-te)#interface gigabitEthernet 0/0/0/3
RP/0/0/CPU0:SP1R1(config-mpls-te)#interface gigabitEthernet 0/0/0/5

! please note we're enabling MPLS-TE to monitor only AS3356 facing links!
```

That should be repeated for SP1R2, SP1R3 and SP1R4.

Before we check the ORR, let's see what new information does OSPF and MPLS TE provide for the link information.

For OSPF it's crucial to check:

```
RP/0/0/CPU0:SP1R1#show ospf AS3356 database

            OSPF Router with ID (33.56.1.1) (Process ID AS3356)
 […]
            Type-10 Opaque Link Area Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Opaque ID
1.0.0.0         33.56.1.1       264         0x80000002 0x001664         0
1.0.0.0         33.56.2.2       327         0x80000002 0x001c5a         0
1.0.0.0         33.56.3.3       274         0x80000002 0x002250         0
1.0.0.0         33.56.4.4       266         0x80000002 0x002846         0
1.0.0.3         33.56.4.4       266         0x80000002 0x008c90         3
1.0.0.4         33.56.1.1       264         0x80000002 0x0043d8         4
1.0.0.4         33.56.2.2       327         0x80000002 0x0057b2         4
1.0.0.4         33.56.3.3       274         0x80000002 0x0017f2         4
1.0.0.5         33.56.4.4       266         0x80000002 0x0048a4         5
1.0.0.6         33.56.1.1       264         0x80000002 0x00081c         6
1.0.0.6         33.56.2.2       327         0x80000002 0x00c75c         6
1.0.0.6         33.56.3.3       274         0x80000002 0x007e6d         6
1.0.0.8         33.56.2.2       327         0x80000002 0x00c23e         8
1.0.0.8         33.56.4.4       266         0x80000002 0x0042be         8
```

Whereas for MPLS TE it's worth to check:

```
RP/0/0/CPU0:SP1R1#sh mpls traffic-eng igp-areas

MPLS-TE IGP Areas

Global router-id:         33.56.1.1
Global optical router-id: Not available

OSPF AS3356

    IGP ID:                      33.56.1.1
    TE router ID configured:     0.0.0.0
                 in use:         33.56.1.1
    Connection:                  up

    area 0
        TE index: 0
        IGP config for TE: complete
        Local links flooded in this IGP area: 2
        Link adjacency IDT: received
        Topology IDT received: received
        P2P tunnel heads running over this IGP area: 0
            0 AA, 0 FA
        P2MP destinations running over this IGP area: 0
        Tunnel loose-hops expanded over this IGP area: 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Now, the ORR SPF will run and propose a topology:

```
RP/0/0/CPU0:SP1R4#show orrspf database detail

ORR policy: SP1R3, IPv4, RIB tableid: 0xe0000012
Configured root: primary: 33.56.3.3, secondary: NULL, tertiary: NULL
Actual Root: 33.56.3.3, Root node: 33.56.3.3

Prefix                                          Cost
33.56.1.1                                       3
33.56.2.2                                       2
33.56.3.3                                       1
33.56.4.4                                       2

Number of mapping entries: 5
```

Let's look at it from SP1R3 perspective:

```
RP/0/0/CPU0:SP1R3#sh bgp ipv4 unicast
[…]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 12.39.0.0/16       22.22.33.254                           0 1239 i
*> 33.56.0.0/16       0.0.0.0                            32768 i
* i                   33.56.4.4                   100        0 i
s> 33.56.23.0/24      0.0.0.0                  0         32768 ?
s> 33.56.34.0/24      0.0.0.0                  0         32768 ?
s> 33.56.35.0/24      0.0.0.0                  0         32768 ?
*>i44.44.67.0/24      33.56.2.2                0    100      0 100 i
*> 66.66.67.0/24      22.22.33.254                           0 1239 200 i
*>i172.16.6.6/32      33.56.2.2                       100      0 100 i
*>i172.16.7.7/32      33.56.2.2                0    100      0 100 i
*> 192.168.6.6/32     22.22.33.254                           0 1239 200 i
*> 192.168.7.7/32     22.22.33.254                           0 1239 200 i

Processed 11 prefixes, 12 paths
```

The AS100 prefixes are now visible via SP1R2 – 33.56.2.2. Let's trace:

```
RP/0/0/CPU0:SP1R3#traceroute 172.16.7.7 source 33.56.3.3
Type escape sequence to abort.
Tracing the route to 172.16.7.7

 1  33.56.23.2 0 msec  0 msec  0 msec
 2  40.60.27.7 0 msec  *  0 msec
```

For such small network, ORR may not look like life-saver, but NSP are already deploying the ORR along with AddPath capabilities in their networks today.

Please roll back to initial configuration before continuing with the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Advanced manipulation of path attributes

Task 2.1: Accumulated IGP cost attribute

One of the challenges in larger scale BGP systems is properly reflecting the traffic engineering requirements within BGP running on underlay IGP.

While IGP metric can be redistributed to BGP by means of metric/MED, the point where MED matters in the BGP best path process is low in the list:

1. (Weight)
2. Highest Local-Preference
3. Shortest AS_PATH
4. Lowest origin code (IGP/EGP/incomplete)
5. Lowest MED

With AIGP, non-transitive BGP path attribute, the best path selection was changed to this:

1. (Weight)
2. Highest Local-Preference
3. Lowest AIGP Cost
4. Shortest AS_PATH
5. Lowest origin code (IGP/EGP/incomplete)
6. Lowest MED

We'll config AS100 routers to send and receive AIGP information:

```
CEL7(config)#router bgp 100
CEL7(config-router)#address-family ipv4 unicast
CEL7(config-router-af)#neighbor 40.60.27.2 aigp
CEL7(config-router-af)#neighbor 172.16.6.6 aigp
CEL7(config-router-af)#address-family ipv6 unicast
CEL7(config-router-af)#neighbor 2001:44:66:CC::6 aigp
CEL7(config-router-af)#neighbor 2001:40:60:27::2 aigp

CEL6(config)#router bgp 100
CEL6(config-router)#address-family ipv4 unicast
CEL6(config-router-af)#neighbor 41.61.16.1 aigp
CEL6(config-router-af)#neighbor 172.16.7.7 aigp
CEL6(config-router)#address-family ipv6 unicast
CEL6(config-router-af)#neighbor 2001:44:66:CC::7 aigp
CEL6(config-router-af)#neighbor 2001:41:61:16::1 aigp
```

We'll also make sure, that IGP metric will be copied during redistribution to AIGP attribute directly from IGP metric by modifying the route-map responsible for redistribution of both IPv4 and IPv6 prefixes:

```
CEL7(config)#route-map CONN->BGP permit 10
CEL7(config-route-map)#set aigp-metric igp-metric
CEL7(config-route-map)#route-map CONN->BGP permit 20
CEL7(config-route-map)#set aigp-metric igp-metric
```

We'll also configure SP1R2 to send AIGP metrics:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#aigp
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#nei 33.56.5.5
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#aigp
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#nei 2001:33:56:aa::5
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#aigp
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#neighbor 2001:40:60:27::7
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#aigp
```

From SP1R2 perspective, path to CEL6 router will lead through eBGP peering, but with accumulated IGP cost of 301:

```
RP/0/0/CPU0:SP1R2#sh ip bgp 172.16.6.6
[…]
Paths: (2 available, best #2)
  Advertised to peers (in unique update groups):
    33.56.5.5
  Path #1: Received by speaker 0
  Not advertised to any peer
  100
    33.56.1.1 (metric 2) from 33.56.5.5 (33.56.1.1)
      Origin IGP, metric 0, localpref 100, valid, internal
      Received Path ID 0, Local Path ID 0, version 0
      Originator: 33.56.1.1, Cluster list: 33.56.5.5
  Path #2: Received by speaker 0
  Advertised to peers (in unique update groups):
    33.56.5.5
  100
    40.60.27.7 from 40.60.27.7 (172.16.7.7)
      Origin IGP, localpref 100, aigp metric 301, valid, external, best, group-best
      Received Path ID 0, Local Path ID 0, version 26
      Origin-AS validity: not-found
      Total AIGP metric 301
```

Please remove all AIGP configuration before moving to the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.2: Using local-as BGP feature

We will change CER7 ASN to 345, but enable it to spoof original ASN 200.

This is current BGP config of CER7:

```
CER7#sh running-config | sect router bgp
router bgp 200
 bgp router-id 192.168.7.7
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 20.40.27.2 remote-as 1239
 neighbor 20.40.27.2 password RIGHT
 neighbor 2001:20:40:27::2 remote-as 1239
 neighbor 2001:20:40:27::2 password RIGHT
 neighbor 2001:22:44:EE::6 remote-as 200
 neighbor 2001:22:44:EE::6 password AS200
 neighbor 192.168.6.6 remote-as 200
 neighbor 192.168.6.6 password AS200
 neighbor 192.168.6.6 update-source Loopback0
 !
 address-family ipv4
  redistribute connected route-map CONN->BGP
  neighbor 20.40.27.2 activate
  neighbor 192.168.6.6 activate
  neighbor 192.168.6.6 next-hop-self
 exit-address-family
 !
 address-family ipv6
  redistribute connected route-map CONN->BGP
  neighbor 2001:20:40:27::2 activate
  neighbor 2001:22:44:EE::6 activate
  neighbor 2001:22:44:EE::6 next-hop-self
 exit-address-family
```

We will change CER7 ASN to 100, but enable it to spoof original ASN 200:

```
CER7(config)#no router bgp 200
CER7(config)#router bgp 100
CER7(config-router)# bgp router-id 192.168.7.7
CER7(config-router)# bgp log-neighbor-changes
CER7(config-router)# no bgp default ipv4-unicast
CER7(config-router)# neighbor 20.40.27.2 remote-as 1239
CER7(config-router)# neighbor 20.40.27.2 password RIGHT
CER7(config-router)# neighbor 20.40.27.2 local-as 200
CER7(config-router)# neighbor 2001:20:40:27::2 remote-as 1239
CER7(config-router)# neighbor 2001:20:40:27::2 password RIGHT
CER7(config-router)# neighbor 2001:20:40:27::2 local-as 200
CER7(config-router)# neighbor 2001:22:44:EE::6 remote-as 200
CER7(config-router)# neighbor 2001:22:44:EE::6 password AS200
CER7(config-router)# neighbor 2001:22:44:EE::6 local-as 200
CER7(config-router)# neighbor 192.168.6.6 remote-as 200
CER7(config-router)# neighbor 192.168.6.6 password AS200
CER7(config-router)# neighbor 192.168.6.6 update-source Loopback0
CER7(config-router)# neighbor 192.168.6.6 local-as 200
CER7(config-router)# address-family ipv4
CER7(config-router-af)#  redistribute connected route-map CONN->BGP
CER7(config-router-af)#  neighbor 20.40.27.2 activate
CER7(config-router-af)#  neighbor 192.168.6.6 activate
CER7(config-router-af)#  neighbor 192.168.6.6 next-hop-self
CER7(config-router-af)# exit-address-family
CER7(config-router)# address-family ipv6
CER7(config-router-af)#  redistribute connected route-map CONN->BGP
CER7(config-router-af)#  neighbor 2001:20:40:27::2 activate
CER7(config-router-af)#  neighbor 2001:22:44:EE::6 activate
CER7(config-router-af)#  neighbor 2001:22:44:EE::6 next-hop-self
CER7(config-router-af)# exit-address-family
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Soon after making changes, sessions will establish like before:

```
CER7#sh bgp ipv4 unicast summary
[…]
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
20.40.27.2      4         1239       7      10        9    0    0 00:02:06        5
192.168.6.6     4          200      10       6        9    0    0 00:02:10        7

CER7#sh bgp ipv6 unicast summary
[…]
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001:20:40:27::2
                4         1239       7      10       11    0    0 00:02:04        5
2001:22:44:EE::6
                4          200      11       7       11    0    0 00:02:11        7
```

Note, that from CER6 (router in the same ASN) everything stays the same:

```
CER6#sh ip bgp neighbors 192.168.7.7 routes
[…]

     Network          Next Hop            Metric LocPrf Weight Path
 * i  66.66.67.0/24    192.168.7.7              0    100      0 i
 r>i  192.168.7.7/32   192.168.7.7              0    100      0 i
```

But for eBGP peers, the AS_PATH will change:

```
RP/0/0/CPU0:SP2R2#sh bgp ipv4 unicast neighbors 20.40.27.7 routes
[…]
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*  66.66.67.0/24      20.40.27.7               0             0 200 100 i
*  192.168.6.6/32     20.40.27.7                             0 200 100 i
*  192.168.7.7/32     20.40.27.7               0             0 200 100 i

Processed 3 prefixes, 3 paths
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.3: Using no-prepend BGP feature

We enabled spoofing of AS 200 by BGP AS 100 process on CER7 in Task 2.2, however, the consequence of the local-as configuration was that it's visible in the AS_PATH from eBGP point of view:

```
RP/0/0/CPU0:SP2R2#sh bgp ipv4 unicast neighbors 20.40.27.7 routes
[…]
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*  66.66.67.0/24      20.40.27.7               0             0 200 100 i
*  192.168.6.6/32     20.40.27.7                             0 200 100 i
*  192.168.7.7/32     20.40.27.7               0             0 200 100 i

Processed 3 prefixes, 3 paths
```

…and from CER7 perspective, all AS_PATHs for prefixes external to AS100 will be prepended by local-as configured value – 200. Let's check this for external prefix 12.39.0.0/16:

```
CER7#sh ip bgp 12.39.0.0/16
BGP routing table entry for 12.39.0.0/16, version 8
Paths: (2 available, best #1, table default)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  1239, (aggregated by 1239 12.39.1.1)
    192.168.6.6 (metric 2) from 192.168.6.6 (192.168.6.6)
      Origin IGP, metric 0, localpref 100, valid, internal, atomic-aggregate, best
      rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 1
  200 1239, (aggregated by 1239 12.39.2.2)
    20.40.27.2 from 20.40.27.2 (12.39.2.2)
      Origin IGP, localpref 100, valid, external, atomic-aggregate
      rx pathid: 0, tx pathid: 0
```

To avoid this from happening, you need to add no-prepend to the local-as configuration. Let's do that on CER7:

```
CER7(config)#router bgp 100
CER7(config-router)#neighbor 20.40.27.2 local-as 200 no-prepend
CER7(config-router)#neighbor 2001:20:40:27::2 local-as 200 no-prepend
```

Once that's done, session will reset and once back, here how the view will change:

```
CER7#sh ip bgp 12.39.0.0/16
BGP routing table entry for 12.39.0.0/16, version 21
Paths: (2 available, best #1, table default)
  Advertised to update-groups:
     2
  Refresh Epoch 1
  1239, (aggregated by 1239 12.39.2.2)
    20.40.27.2 from 20.40.27.2 (12.39.2.2)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, best
      rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 1
  1239, (aggregated by 1239 12.39.1.1)
    192.168.6.6 (metric 2) from 192.168.6.6 (192.168.6.6)
      Origin IGP, metric 0, localpref 100, valid, internal, atomic-aggregate
      rx pathid: 0, tx pathid: 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.4: Using replace-as BGP feature

Once we have the configuration from Task 2.3 in place, we're still visible in AS_PATH from external ASNs. To fully replace original ASN with the local-as configured one, you need to do both no-prepend and replace-as at the same time like this:

```
CER7#conf t
CER7(config)#router bgp 100
CER7(config-router)#neighbor 20.40.27.2 local-as 200 no-prepend replace-as
CER7(config-router)#neighbor 2001:20:40:27::2 local-as 200 no-prepend replace-as
CER7(config-router)#^Z
```

Now, the AS 200 is the only AS_PATH attribute from external point of view and we no longer see "original" AS 100:

```
RP/0/0/CPU0:SP2R2#sh bgp ipv4 unicast neighbors 20.40.27.7 routes
Sat Jan  1 22:41:58.974 UTC
BGP router identifier 12.39.2.2, local AS number 1239
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 57
BGP main routing table version 57
BGP NSR Initial initsync version 5 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 20.40.17.0/24      20.40.27.7               0             0 200 i
*> 20.40.27.0/24      20.40.27.7               0             0 200 i
*> 21.41.16.0/24      20.40.27.7                             0 200 i
*> 21.41.26.0/24      20.40.27.7                             0 200 i
*> 66.66.67.0/24      20.40.27.7               0             0 200 i
*> 192.168.6.6/32     20.40.27.7                             0 200 i
*> 192.168.7.7/32     20.40.27.7               0             0 200 i

Processed 7 prefixes, 7 paths
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.5: Using dual-as BGP feature

If you want to let external routers establish sessions from both original and spoofed ASN, you need to enable dual-as feature on neighbor:

```
CER7#conf t
CER7(config)#router bgp 100
CER7(config-router)#neighbor 20.40.27.2 local-as 200 no-prepend replace-as dual-as
CER7(config-router)#neighbor 2001:20:40:27::2 local-as 200 no-prepend replace-as dual-as
```

Note dual-as requires use of no-prepend and replace-as at the same time. Once it's done, the sessions will reestablish:

```
RP/0/0/CPU0:SP2R2#sh bgp ipv4 unicast summary | b Neighbor
Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
12.39.5.5         0  1239      44      41       85    0    0 00:29:34          16
20.40.27.7        0   200     103      77       85    0    0 00:01:24           7
```

Prefixes advertised by 20.40.27.7 are of course received with AS_PATH equal to 200 only – because of the no-prepend and replace-as config:

```
RP/0/0/CPU0:SP2R2#sh bgp ipv4 unicast neighbors 20.40.27.7 routes
[…]
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 20.40.17.0/24      20.40.27.7               0             0 200 i
*> 20.40.27.0/24      20.40.27.7               0             0 200 i
*> 21.41.16.0/24      20.40.27.7                             0 200 i
*> 21.41.26.0/24      20.40.27.7                             0 200 i
*> 66.66.67.0/24      20.40.27.7               0             0 200 i
*> 192.168.6.6/32     20.40.27.7                             0 200 i
*> 192.168.7.7/32     20.40.27.7               0             0 200 i
```

Now, let's reconfigure SP2R2 session to 20.40.27.7 to establish a session with AS100 instead of the original AS200:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#router bgp 1239
RP/0/0/CPU0:SP2R2(config-bgp)#neighbor 20.40.27.7
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#remote-as 100
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#neighbor 2001:20:40:27::7
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#remote-as 100
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#commit
```

Now the information should point to ASN 100 and session should be alive:

```
RP/0/0/CPU0:SP2R2#sh bgp ipv4 unicast summary | b Neighbor
Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
12.39.5.5         0  1239      48      47       99    0    0 00:33:06          16
20.40.27.7        0   100     115      86       99    0    0 00:00:27           7
```

…and:

```
RP/0/0/CPU0:SP2R2#sh bgp ipv4 unicast neighbors 20.40.27.7 routes
[…]
   Network            Next Hop            Metric LocPrf Weight Path
*> 20.40.17.0/24      20.40.27.7               0             0 100 i
*> 20.40.27.0/24      20.40.27.7               0             0 100 i
*> 21.41.16.0/24      20.40.27.7                             0 100 i
*> 21.41.26.0/24      20.40.27.7                             0 100 i
*> 66.66.67.0/24      20.40.27.7               0             0 100 i
*> 192.168.6.6/32     20.40.27.7                             0 100 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
*> 192.168.7.7/32     20.40.27.7               0             0 100 i

Processed 7 prefixes, 7 paths
```

So this demonstrates that SP2R2 can now establish sessions to R7 using both ASN200 and AS100.

This concludes spoofing of different ASNs that can be useful for migrations. Please get back to default configuration of R7 and SP2R2 (to config from Task 2.1).

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.6: Using allowas-in BGP feature

To check capability of accepting own AS in the path, let's shut down the neighborship between SP2R1 and CER6:

```
RP/0/0/CPU0:SP2R1(config)#router bgp 1239
RP/0/0/CPU0:SP2R1(config-bgp)#neighbor 21.41.16.6
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#shut
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#
```

This will assure us, that the path via SP2R2 is the best path and will be propagated across AS1239 to AS3356 and to AS100 on the left side.

On the SP2R3:

```
RP/0/0/CPU0:SP2R3#sh ip bgp 66.66.67.0
[…]
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    22.22.33.1
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    22.22.33.1
  200 100
    12.39.2.2 (metric 20) from 12.39.5.5 (12.39.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal, best, group-best
      Received Path ID 0, Local Path ID 0, version 222
      Originator: 12.39.2.2, Cluster list: 12.39.5.5
```

On the SP1R1:

```
RP/0/0/CPU0:SP1R1#sh ip bgp 66.66.67.0
[…]
Paths: (2 available, best #1)
  Advertised to peers (in unique update groups):
    41.61.16.6
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    41.61.16.6
  1239 200 100
    33.56.3.3 (metric 2) from 33.56.5.5 (33.56.3.3)
      Origin IGP, localpref 100, valid, internal, best, group-best, multipath
      Received Path ID 0, Local Path ID 0, version 611
      Originator: 33.56.3.3, Cluster list: 33.56.5.5
  Path #2: Received by speaker 0
  Not advertised to any peer
  1239 200 100
    33.56.4.4 (metric 2) from 33.56.5.5 (33.56.4.4)
      Origin IGP, localpref 100, valid, internal, multipath, add-path
      Received Path ID 1, Local Path ID 2, version 611
      Originator: 33.56.4.4, Cluster list: 33.56.5.5
```

However, from both CEL6 and CEL7 the route vanished, as both routers detected AS_PATH loop:

```
BGP(0): 40.60.27.2 rcv UPDATE about 192.168.7.7/32 -- DENIED due to: AS-PATH contains
our own AS;
BGP(0): 40.60.27.2 rcv UPDATE about 192.168.6.6/32 -- DENIED due to: AS-PATH contains
our own AS;
BGP(0): 40.60.27.2 rcv UPDATE about 66.66.67.0/24 -- DENIED due to: AS-PATH contains our
own AS;
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

We will reconfigure both CEL6 and CEL7 to allow for their own ASN to appear in the AS_PATH:

```
CEL6(config)#router bgp 100
CEL6(config-router)#address-family ipv4 unicast
CEL6(config-router-af)#neighbor 41.61.16.1 allowas-in

CEL7(config)#router bgp 100
CEL7(config-router)#address-family ipv4 unicast
CEL7(config-router-af)#neighbor 40.60.27.2 allowas-in
```

Now, despite having their own ASN appearing as origin for the prefixes, both CEL6 and CEL7 happily accept the prefixes.

For CEL6:

```
CEL6#sh ip bgp
[…]

     Network          Next Hop            Metric LocPrf Weight Path
 * i  12.39.0.0/16     172.16.7.7               0    100      0 3356 1239 i
 *>                    41.61.16.1                            0 3356 1239 i
 * i  33.56.0.0/16     172.16.7.7               0    100      0 3356 i
 *>                    41.61.16.1                            0 3356 i
 * i  44.44.67.0/24    172.16.7.7               0    100      0 i
 *>                    0.0.0.0                  0         32768 i
 * i  66.66.67.0/24    172.16.7.7               0    100      0 3356 1239 200 100 i
 *>                    41.61.16.1                            0 3356 1239 200 100 i
 *>   172.16.6.6/32    0.0.0.0                  0         32768 i
 r>i  172.16.7.7/32    172.16.7.7               0    100      0 i
 * i  192.168.6.6/32   172.16.7.7               0    100      0 3356 1239 200 100 i
 *>                    41.61.16.1                            0 3356 1239 200 100 i
 * i  192.168.7.7/32   172.16.7.7               0    100      0 3356 1239 200 100 i
 *>                    41.61.16.1                            0 3356 1239 200 100 i
```

And CEL7:

```
CEL7#sh ip bgp
[…]

     Network          Next Hop            Metric LocPrf Weight Path
 *>   12.39.0.0/16     40.60.27.2                            0 3356 1239 i
 * i                   172.16.6.6               0    100      0 3356 1239 i
 *>   33.56.0.0/16     40.60.27.2                            0 3356 i
 * i                   172.16.6.6               0    100      0 3356 i
 *>   44.44.67.0/24    0.0.0.0                  0         32768 i
 * i                   172.16.6.6               0    100      0 i
 *>   66.66.67.0/24    40.60.27.2                            0 3356 1239 200 100 i
 * i                   172.16.6.6               0    100      0 3356 1239 200 100 i
 r>i  172.16.6.6/32    172.16.6.6               0    100      0 i
 *>   172.16.7.7/32    0.0.0.0                  0         32768 i
 *>   192.168.6.6/32   40.60.27.2                            0 3356 1239 200 100 i
 * i                   172.16.6.6               0    100      0 3356 1239 200 100 i
 *>   192.168.7.7/32   40.60.27.2                            0 3356 1239 200 100 i
 * i                   172.16.6.6               0    100      0 3356 1239 200 100 i
```

Please roll back the Task changes before moving to the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.7: Using remove-private-as BGP feature

The feature works only on outbound eBGP sessions. To simulate an AS distributing private ASNs to “global” BGP table, we’ll use RPL and AS3356 as initiator.

```
RP/0/0/CPU0:SP1R2(config)#route-policy SET-ASPATH
RP/0/0/CPU0:SP1R2(config-rpl)#prepend as-path 65055 3
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
```

Now, we’ll build another policy, that apart from inheriting the SET-ASPATH policy will simply ‘pass’ all prefixes towards AS1239 from AS3356:

```
RP/0/0/CPU0:SP1R4(config)#route-policy AS3356-MOD
RP/0/0/CPU0:SP1R4(config-rpl)#apply SET-ASPATH
RP/0/0/CPU0:SP1R4(config-rpl)#pass
RP/0/0/CPU0:SP1R4(config-rpl)#end-policy
```

Now, let’s apply the new route-policy to outbound towards AS1239 SP2R4 and clear the session:

```
RP/0/0/CPU0:SP1R4(config)#router bgp 3356
RP/0/0/CPU0:SP1R4(config-bgp)#neighbor 11.11.11.254
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R4(config-bgp-nbr-af)#route-policy AS3356-MOD out
RP/0/0/CPU0:SP1R4(config-bgp-nbr-af)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
RP/0/0/CPU0:SP1R4#clear ip bgp 11.11.11.254
```

SP2R4 now has all prefixes from AS3356 advertised via SP1R4 modified:

```
RP/0/0/CPU0:SP2R4#sh ip bgp
[…]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 12.39.0.0/16       0.0.0.0                            32768 i
* i                   12.39.1.1                     100      0 i
s> 12.39.14.0/24      0.0.0.0                  0         32768 ?
s> 12.39.24.0/24      0.0.0.0                  0         32768 ?
s> 12.39.34.0/24      0.0.0.0                  0         32768 ?
*  33.56.0.0/16       11.11.11.1                             0 3356 65055 65055 65055 i
*>i                   12.39.3.3                     100      0 3356 i
*  44.44.67.0/24      11.11.11.1                             0 3356 65055 65055 65055
100 i
*>i                   12.39.3.3                     100      0 3356 100 i
*>i66.66.67.0/24      12.39.2.2                0    100      0 200 100 i
*  172.16.6.6/32      11.11.11.1                             0 3356 65055 65055 65055
100 i
*>i                   12.39.3.3                     100      0 3356 100 i
*  172.16.7.7/32      11.11.11.1                             0 3356 65055 65055 65055
100 i
*>i                   12.39.3.3                     100      0 3356 100 i
*>i192.168.6.6/32     12.39.2.2                     100      0 200 100 i
*>i192.168.7.7/32     12.39.2.2                0    100      0 200 100 i

Processed 11 prefixes, 16 paths
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

To force the paths with longer AS_PATH propagate in AS1239 we'll artificially increase LOCAL_PREF to 5000 on routers received by SP2R4:

```
RP/0/0/CPU0:SP2R4(config)#route-policy SET-LP
RP/0/0/CPU0:SP2R4(config-rpl)#set local-preference 5000
RP/0/0/CPU0:SP2R4(config-rpl)#pass
RP/0/0/CPU0:SP2R4(config-rpl)#end-policy
```

Now, let's attach the policy to neighbor:

```
RP/0/0/CPU0:SP2R4(config)#router bgp 1239
RP/0/0/CPU0:SP2R4(config-bgp)#neighbor 11.11.11.1
RP/0/0/CPU0:SP2R4(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R4(config-bgp-nbr-af)#route-policy SET-LP in
RP/0/0/CPU0:SP2R4(config-bgp-nbr-af)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

On CER7 we can see at least one path that includes Private ASNs propagated via AS1239:

```
CER7#sh ip bgp 33.56.0.0
BGP routing table entry for 33.56.0.0/16, version 15
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  200 1239 3356 65055 65055 65055, (aggregated by 3356 33.56.4.4)
    20.40.27.2 from 20.40.27.2 (12.39.2.2)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, best
      rx pathid: 0, tx pathid: 0x0
```

Let's fix it by removing private ASNs from AS_PATH outbound on SP2R2:

```
RP/0/0/CPU0:SP2R2(config)#router bgp 1239
RP/0/0/CPU0:SP2R2(config-bgp)#neighbor 20.40.27.7
RP/0/0/CPU0:SP2R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R2(config-bgp-nbr-af)#remove-private-AS
RP/0/0/CPU0:SP2R2(config-bgp-nbr-af)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

The AS_PATH is now clean:

```
CER7#sh ip bgp 33.56.0.0
BGP routing table entry for 33.56.0.0/16, version 16
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  200 1239 3356, (aggregated by 3356 33.56.4.4)
    20.40.27.2 from 20.40.27.2 (12.39.2.2)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, best
      rx pathid: 0, tx pathid: 0x0
```

It's worth to note, that IOS XR has also ability to filter the private ASNs from eBGP neighbor also on **inbound**:

```
RP/0/0/CPU0:SP2R2(config-bgp-nbr-af)#remove-private-AS ?
  entire-aspath        remove only if all ASes in the path are private
  inbound              Remove private AS number from inbound updates
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Before continuing to the next task, please unshut connection between AS1239 and AS200:

```
RP/0/0/CPU0:SP2R1#conf t
RP/0/0/CPU0:SP2R1(config)#router bgp 1239
RP/0/0/CPU0:SP2R1(config-bgp)#neighbor 21.41.16.6
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#no shut
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#
```

Also, please roll back other Task changes before moving to the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.8: Manipulating communities

Most of the signalling between different ASes happens by means of communities. Manipulating them is one of the most common tasks in live SP networks.

Let's go through typical community manipulations.

Overwriting communities via route-map on IOS/IOS-XE:

```
CER7(config)#route-map TEST permit 10
CER7(config-route-map)#set community 333:100

! note this resets ALL existing community on matched object
! as this route-map doesn't specify 'match' – it'll reset community on
! all prefixes it's applied to
```

Adding communities via route-map on IOS/IOS-XE:

```
CER7(config)#route-map TEST permit 10
CER7(config-route-map)#set community 333:100 additive

! note this will add to existing community set one with value of 333:100
```

Deleting specific community via route-map on IOS/IOS-XE:

```
CEL7(config)#ip community-list 10 permit 333:100

CEL7(config)#route-map COMM-DELETE permit 10
CEL7(config-route-map)#set comm-list 10 delete

! if community exists for specific processed prefix, it will be
! removed (but only this out of potential whole string of communities
! assigned to prefix)

CEL7(config)#route-map COMM-DELETE permit 20

! all other prefixes and their communities will be left alone
! and accepted
```

Overwriting communities via RPL in IOS XR:

```
RP/0/0/CPU0:SP2R1(config-rpl)#set community ( no-export )
```

Adding communities via route-map on IOS XR:

```
RP/0/0/CPU0:SP2R1(config-rpl)#set community ( no-export ) additive
```

Editing community within RPL – deleting specific community:

```
RP/0/0/CPU0:SP2R1(config)#route-policy MAN-COMM
RP/0/0/CPU0:SP2R1(config-rpl)#delete community in ( 100:100 )
```

Please roll back Task changes before moving to the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.9: Attribute filtering

For BGP neighbor sessions, you may be interested in filtering out unsupported/problematic attributes.

For IOS/IOS-XE the feature is:

```
CER6(config)#router bgp 200
CER6(config-router)#neighbor 21.41.16.1 path-attribute discard range 30 50 in
```

This example filters attributes in range between 30 and 50.

While you don't have full freedom of filtering all attributes (as it could threaten the BGP mechanics), you can filter specific 'feature' attributes.



Filtering BGP attributes on sessions with external peers is serious interference that can impact the stability of your network and your transit traffic. Such advanced operation should be verified before deploying in live network. In lab networks of course, everything is fair call.

IANA keeps track of official attribute mappings:
https://www.iana.org/assignments/bgp-parameters/bgp-parameters.xhtml#bgp-parameters-2

And please also note, that there ARE systems in global BGP table that advertise vendor-specific information using attributes nobody cared to register at IANA.

For example, recently adopted RFC 8093 calls for deprecation of BGP Path Attributes of 30, 31, 129, 241, 242 and 243. Let's do exactly that:

```
CER6(config-router)#neighbor 21.41.16.1 path-attribute discard 30 in
CER6(config-router)#neighbor 21.41.16.1 path-attribute discard 31 in
CER6(config-router)#neighbor 21.41.16.1 path-attribute discard 129 in
CER6(config-router)#neighbor 21.41.16.1 path-attribute discard 241 in
CER6(config-router)#neighbor 21.41.16.1 path-attribute discard 242 in
CER6(config-router)#neighbor 21.41.16.1 path-attribute discard 243 in
```

There's also another option – treating the attribution advertisement as explicit withdraw:

```
CER6(config-router)#neighbor 21.41.16.1 path-attribute treat-as-withdraw 240 in
```

On IOS XR the filtering is applied using attribute-filtering groups in this manner:

```
RP/0/0/CPU0:SP2R1(config)#router bgp 1239
RP/0/0/CPU0:SP2R1(config-bgp)#attribute-filter group AF_GROUP
RP/0/0/CPU0:SP2R1(config-bgp-attrfg)#attribute ?
  0                    Attribute code
  <16-255>             Attribute code
  <4-13>               Attribute code
  AGGREGATOR           Aggregator attribute (7)
  AIGP                 AIGP attribute (26)
  AS4-AGGREGATOR       AS4_AGGREGATOR attribute (18)
  AS4-PATH             AS4_PATH attribute (17)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  ATOMIC-AGGREGATE      Atomic aggrgate attribute (6)
  ATTRSET               ATTR_SET attribute (128)
  CLUSTER-LIST          Cluster List attribute (10)
  COMMUNITY             Community attribute (8)
  CONNECTOR             Connector attribute (20)
  EXTENDED-COMMUNITIES  Extended community attribute (16)
  LOCAL-PREF            Local Preference attribute (5)
  LSATTR                Link-state attribute (99)
  MULTI-EXIT-DISC       MED attribute (4)
  ORIGINATOR-ID         Originator attribute (9)
  PMSI-TUNNEL           PMSI Tunnel attribute (22)
  PPMP                  PPMP attribute (70)
  SSA                   SAFI specific attribute (19)
  range                 Attribute code range
  unrecognized          All unrecognized attributes
RP/0/0/CPU0:SP2R1(config-bgp-attrfg)#attribute unrecognized ?
  discard             Discard this attribute
  treat-as-withdraw   Treat the update message as a Withdraw

RP/0/0/CPU0:SP2R1(config-bgp)#neighbor 12.39.5.5
RP/0/0/CPU0:SP2R1(config-bgp-nbr)#update in filtering attribute-filter AF_GROUP
```

Please roll back Task changes before moving to the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.10: Selective BGP AD change for prefix

IOS XR enables us to selectively change administrative distance for specific prefix or group of prefixes. This feature may make sense in situations, where you want a specific prefix to be inserted in routing table via BGP and not via any other protocol – like any of the IGPs if we're talking about iBGP provided prefixes.

Configuration uses global per-AF table-policy feature. Let's find a prefix that we will change AD for – while your topology and router may differ, just find any eBGP provided prefix.

We'll use SP1R1 and 172.16.6.6/32:

```
RP/0/0/CPU0:SP1R1#sh ip bgp
BGP router identifier 33.56.1.1, local AS number 3356
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 29
BGP main routing table version 29
BGP NSR Initial initsync version 3 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
[…]
*> 172.16.6.6/32      41.61.16.6               0             0 100 i
```

Let's check what's the router routing table showing for this prefix:

```
RP/0/0/CPU0:SP1R1#sh ip route 172.16.6.6

Routing entry for 172.16.6.6/32
  Known via "bgp 3356", distance 20, metric 0
  Tag 100, type external
  Installed Jan  1 22:21:01.436 for 1w0d
  Routing Descriptor Blocks
    41.61.16.6, from 41.61.16.6, BGP external
      Route metric is 0
  No advertising protos.
```

So, the prefix is in both BGP and routing table, and AD distance is 20. We'll now build policy matching this prefix and adjusting AD:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#route-policy BGP-SAD
RP/0/0/CPU0:SP1R1(config-rpl)#if destination in ( 172.16.6.6/32 ) then
RP/0/0/CPU0:SP1R1(config-rpl-if)#    set administrative-distance 5
RP/0/0/CPU0:SP1R1(config-rpl-if)#  else
RP/0/0/CPU0:SP1R1(config-rpl-else)#    pass
RP/0/0/CPU0:SP1R1(config-rpl-else)#  endif
RP/0/0/CPU0:SP1R1(config-rpl)#end-policy
RP/0/0/CPU0:SP1R1(config)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

We need to attach it to AF IPv4 unicast (or other specific AF you want to adjust AD for):

```
RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-af)#table-policy ?
  PASS      Name of the policy
  BGP-SAD   Name of the policy
  RED-BGP   Name of the policy
  WORD      Name of the policy
RP/0/0/CPU0:SP1R1(config-bgp-af)#table-policy BGP-SAD
RP/0/0/CPU0:SP1R1(config-bgp-af)#commit
```

Let's check the current view:

```
RP/0/0/CPU0:SP1R1#sh route
[…]

Gateway of last resort is not set

B    66.66.67.0/24 [200/0] via 33.56.3.3, 00:00:05
B    172.16.6.6/32 [5/0] via 41.61.16.6, 00:00:05
B    172.16.7.7/32 [20/0] via 41.61.16.6, 00:00:05
B    192.168.6.6/32 [200/0] via 33.56.3.3, 00:00:05
B    192.168.7.7/32 [200/0] via 33.56.3.3, 00:00:05
```

You can see that we have multiple eBGP and iBGP provided prefixes in the routing table, however for 172.16.6.6/32 the AD was changed to 5 vs 20 for default eBGP and 200 for iBGP.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3: Real world BGP ISP challenges

Task 3.1: BGP route dampening

Now let's configure BGP dampening on AS3356 and flap AS100 (CEL6) networks multiple times.

Also, note – we can limit (using route-policy) set of networks we're interested about in terms of applying flapping penalties.

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-af)#bgp dampening
RP/0/0/CPU0:SP1R2(config-bgp-af)#
RP/0/0/CPU0:SP1R2(config-bgp)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R2(config-bgp-af)#bgp dampening

Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```



With an introduction of very powerful Route Processors (based mainly on Intel x86 platforms), the BGP route damping was causing more reachability problems than it solved and at some point in 2005-2006 considered harmful. While core networks should be free from BGP damping, edge connections, prone to reconfigurations and/or errors may benefit from additional 'safety' of this mechanism. RIPE (European RIR) officially discouraged use of BGP damping back in 2006, but recently decided it may be of use in some scenarios with tuned weights and timers. That is in line with RFC 7196 that was baseline for the recommendation:
https://www.ripe.net/publications/docs/ripe-580
RIPE 229 (dated back to 2001!) recommended BGP damping but exclude routes leading to critical internet services ("golden networks"), like DNS root servers.

After one "**clear ip bgp ***" on CEL6 we see on SP1R2 (route not yet dampened):

```
RP/0/0/CPU0:SP1R2#show bgp flap-statistics 172.16.6.6 detail
[…]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            From            Flaps Duration Reuse    Path
* i172.16.6.6/32      33.56.1.1       0                       100 i

   Half life        Suppress        Reuse penalty    Max. supp. time
   00:15:00         2000            750              01:00:00

*                     40.60.27.7       1     00:01:39          100 i

   Half life        Suppress        Reuse penalty    Max. supp. time
   00:15:00         2000            750              01:00:00

*>                    41.61.26.6       0                       100 i

   Half life        Suppress        Reuse penalty    Max. supp. time
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
   00:15:00         2000            750              01:00:00
```

Let's flap the route twice from CEL6. Route will move in 'history' state and SP1R2 will stop announcing it to its peers – the SP1R5 (RR) in this case:

```
RP/0/0/CPU0:SP1R2#show bgp flap-statistics 172.16.6.6 detail
[…]
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            From            Flaps Duration Reuse    Path
 h 172.16.6.6/32      40.60.27.7      3     00:20:03          100 i

   Half life        Suppress        Reuse penalty    Max. supp. time
   00:15:00         2000            750              01:00:00

RP/0/0/CPU0:SP1R2#sh ip bgp 172.16.6.6
[…]
Paths: (1 available, no best path)
  Advertised to peers (in unique update groups):
    40.60.27.7
  Path #1: Received by speaker 0
  Not advertised to any peer
  100, (history entry)
    40.60.27.7 from 40.60.27.7 (172.16.7.7)
      Origin IGP, localpref 100, external
      Received Path ID 0, Local Path ID 0, version 0
      Origin-AS validity: not-found
```

Finally, with next flap the route will move into '**d**' for '**dampen**' state:

```
RP/0/0/CPU0:SP1R2#show bgp flap-statistics 172.16.6.6 detail
[…]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            From            Flaps Duration Reuse    Path
[…]
*d                    40.60.27.7      4     00:24:21 00:25:20 100 i

   Half life        Suppress        Reuse penalty    Max. supp. time
   00:15:00         2000            750              01:00:00

*>                    41.61.26.6      0                        100 i

   Half life        Suppress        Reuse penalty    Max. supp. time
   00:15:00         2000            750              01:00:00
```

In this situation, the '**reuse**' timer needs to clock down from the **25m20s** for the route to be usable again.

By using '**show bgp dampened-paths**' you can check the situation in much simpler manner:

```
RP/0/0/CPU0:SP1R2#sh bgp dampened-paths
[…]
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            From            Reuse    Path
*d 172.16.6.6/32      40.60.27.7      00:25:18 100 i
```

Please roll back Task changes before moving to the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2: Selective FIB download – IOS/IOS-XE

Sometimes you can't or don't want to fit entire RIB in FIB. Ability to do so is in both IOS-XE and IOS XR.

Let's try to check if we can force router to install only /32 prefixes for IPv4.

We start with following entries in routing table installed by BGP:

```
CEL7#sh ip route | i B
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
B        12.39.0.0 [200/0] via 172.16.6.6, 00:12:57
B        33.56.0.0 [200/0] via 172.16.6.6, 01:17:12
B        66.66.67.0 [200/0] via 172.16.6.6, 00:20:59
B        192.168.6.6 [200/0] via 172.16.6.6, 00:20:59
B        192.168.7.7 [200/0] via 172.16.6.6, 00:20:59
```

Let's define prefix list that will be used by route-map (that, for Selective FIB download is called table-map):

```
CEL7(config)#ip prefix-list SRD-32 permit 0.0.0.0/0 ge 32
CEL7(config)#route-map SRD-32 permit 10
CEL7(config-route-map)#match ip address prefix-list SRD-32
```

Now, let's apply this under appropriate address family:

```
CEL7(config-route-map)#router bgp 100
CEL7(config-router)#address-family ipv4 unicast
CEL7(config-router-af)#table-map SRD-32 filter
```

Now, let's force clearing out existing entries via configured table-map:

```
CEL7#clear ip bgp table-map
```

As you can see, only /32 from BGP are installed in the routing table:

```
CEL7#sh ip route | i B
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
B        192.168.6.6 [200/0] via 172.16.6.6, 00:00:02
B        192.168.7.7 [200/0] via 172.16.6.6, 00:00:02
```

Please remove this feature from CEL7 before moving to next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.3: Selective FIB download – IOS XR

Before SRD configuration – a mix of /16 and /24 installed by BGP:

```
RP/0/0/CPU0:SP1R2#sh ip ro | i B
Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       M - mobile route, r - RPL, (!) - FRR Backup path
B     12.39.0.0/16 [200/0] via 33.56.4.4, 00:02:31
B     33.56.0.0/16 [200/0] via 0.0.0.0, 04:18:53, Null0
B     44.44.67.0/24 [20/0] via 40.60.27.7, 01:02:19
B     66.66.67.0/24 [200/0] via 33.56.4.4, 00:02:31
B     172.16.6.6/32 [20/0] via 40.60.27.7, 01:02:19
B     172.16.7.7/32 [20/0] via 40.60.27.7, 01:02:19
B     192.168.6.6/32 [200/0] via 33.56.4.4, 00:02:31
B     192.168.7.7/32 [200/0] via 33.56.4.4, 00:02:31
```

Now, let's configure route-policy that will filter anything apart from /32 prefixes:

```
RP/0/0/CPU0:SP1R2(config)#route-policy SRD-32
RP/0/0/CPU0:SP1R2(config-rpl)#if destination in ( 0.0.0.0/0 eq 32 ) then
RP/0/0/CPU0:SP1R2(config-rpl-if)#pass
RP/0/0/CPU0:SP1R2(config-rpl-if)#endif
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-af)#table-policy SRD-32
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

Let's again look into routing table with entries from BGP:

```
RP/0/0/CPU0:SP1R2#sh ip ro | i B
Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       M - mobile route, r - RPL, (!) - FRR Backup path
B     172.16.6.6/32 [20/0] via 40.60.27.7, 00:04:25
B     172.16.7.7/32 [20/0] via 40.60.27.7, 00:04:25
B     192.168.6.6/32 [200/0] via 33.56.4.4, 00:04:25
B     192.168.7.7/32 [200/0] via 33.56.4.4, 00:04:25
```

No more anything else than /32s from BGP are present in the routing table.

Please roll back Task changes before moving to the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.4: Multi-instance BGP

As IOS XR was built from the grounds up as fully distributed network operating systems, the capability to run different protocols as different processes was used to provide capabilities in physical routing systems to distribute the load to different route processor cards.

One of the interesting effects is that BGP can be instructed to work in separate instance for specific address-family (note, address families can't overlap between instances). We'll use this capability to migrate BGP process on SP1R1 to two different instances, each one running it's own address family.

BGP instance named "IPv4" will run AF IPv4, while "IPv6" will run AF IPv6. Note, it's recommended to run multicast AF in their respective unicast counterparts to enable proper address and routing resolution.

First, let's delete SP1R1 BGP process AS3356:

```
RP/0/0/CPU0:SP1R1(config)#no router bgp 3356
RP/0/0/CPU0:SP1R1(config)#commit
```

Now, let's configure two instances separating AFs between each of them::

```
RP/0/0/CPU0:SP1R1(config)#router bgp 3356 instance IPv4
RP/0/0/CPU0:SP1R1(config-bgp)# address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-af)#  aggregate-address 33.56.0.0/16 summary-only
RP/0/0/CPU0:SP1R1(config-bgp-af)#  redistribute connected route-policy RED-BGP
RP/0/0/CPU0:SP1R1(config-bgp-af)# neighbor 33.56.5.5
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  remote-as 3356
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  password encrypted 13242441585952
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  update-source Loopback0
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#   next-hop-self
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)# neighbor 41.61.16.6
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  remote-as 100
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  password encrypted 132932343F
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#   route-policy PASS in
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#   route-policy PASS out
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#exit
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#exit
RP/0/0/CPU0:SP1R1(config-bgp)#commit

RP/0/0/CPU0:SP1R1(config-bgp)#router bgp 3356 instance IPv6
RP/0/0/CPU0:SP1R1(config-bgp)# bgp router-id 9.9.9.9

! please note, as per IOS XR BGP router-id selection rules, you NEED to configure
! BGP router-id explicitly if no other loopbacks exist on the router;
! IOS XR will NOT pick the router-id from physical interface by itself

RP/0/0/CPU0:SP1R1(config-bgp)# address-family ipv6 unicast
RP/0/0/CPU0:SP1R1(config-bgp-af)#  aggregate-address 2001:33:56::/48 summary-only
RP/0/0/CPU0:SP1R1(config-bgp-af)#  redistribute connected route-policy RED-BGP
RP/0/0/CPU0:SP1R1(config-bgp-af)# neighbor 2001:33:56:aa::5
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  remote-as 3356
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  password encrypted 02273708585359
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  update-source Loopback0
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  address-family ipv6 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#   next-hop-self
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)# neighbor 2001:41:61:16::6
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  remote-as 100
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  password encrypted 013F23226F
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#  address-family ipv6 unicast
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#   route-policy PASS in
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#    route-policy PASS out
RP/0/0/CPU0:SP1R1(config-bgp-nbr-af)#exit
RP/0/0/CPU0:SP1R1(config-bgp-nbr)#exit
RP/0/0/CPU0:SP1R1(config-bgp)#exit
RP/0/0/CPU0:SP1R1(config)#commit
```

You can verify BGP functioning properly with two separate instances (and processes):

```
RP/0/0/CPU0:SP1R1#show bgp instance all ipv4 unicast summary
BGP instance 0: 'IPv4'
======================
BGP router identifier 33.56.1.1, local AS number 3356
[…]

Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker              20         20         20         20          20           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
33.56.5.5         0  3356      19      12       20    0    0 00:06:58          4
41.61.16.6        0   100      16      12       20    0    0 00:06:57          3


BGP instance 1: 'IPv6'
======================
% None of the requested address families are configured for instance 'IPv6'(26770)
```

And for IPv6 unicast:

```
RP/0/0/CPU0:SP1R1#show bgp instance all ipv6 unicast summary
BGP instance 1: 'IPv6'
======================
BGP router identifier 9.9.9.9, local AS number 3356
[…]

Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker              23         23         23         23          23           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
2001:33:56:aa::5
                  1  3356      17       9       23    0    0 00:02:15          4
2001:41:61:16::6
                  1   100      14       8       23    0    0 00:02:09          3
```

You can use one of the node diagnostic commands to check which BGP process has which JID:

```
RP/0/0/CPU0:SP1R1#show placement program bgp
Display program related information. This is the program information corresponding
to this LR as perceived by the placement daemon.
--------------------------------------------------------------------------------
---
                   Process Information
--------------------------------------------------------------------------------
---
Program         Group           jib  Active         Active-state   Standby     Standby-
state
--------------------------------------------------------------------------------
---
bgp(IPv4)       v4-routing      1053 0/0/CPU0       RUNNING        NONE
NOT_SPAWNED
bgp(IPv6)       Group_10_bgp2   1054 0/0/CPU0       RUNNING        NONE
NOT_SPAWNED
--------------------------------------------------------------------------------
---
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Based on that, you can get process detailed information, including from where it was started, in which mode it's running and what's the state of the process:

```
RP/0/0/CPU0:SP1R1#show processes 1053
                  Job Id: 1053
                     PID: 942370
          Executable path: /disk0/iosxr-routing-6.1.2/bin/bgp
               Instance #: 2
               Version ID: 00.00.0000
                  Respawn: ON
            Respawn count: 1
             Last started: Thu Jan  1 13:18:39 2017
            Process state: Run
            Package state: Normal
       Started on config: IPv4
            Feature name: ON
                     Tag : IPv4
            Process group: v4-routing
                     core: MAINMEM
                Max. core: 0
                Placement: Placeable
             startup_path: /pkg/startup/bgp.startup
                    Ready: 0.749s
                Available: 8.069s
        Process cpu time: 0.060 user, 0.010 kernel, 0.070 total
JID    TID  Stack pri state         TimeInState       HR:MM:SS:MSEC NAME
1053   1     444K  10 Receive         0:00:00:0159    0:00:00:0040 bgp
[…]

RP/0/0/CPU0:SP1R1#show processes 1054
                  Job Id: 1054
                     PID: 942372
          Executable path: /disk0/iosxr-routing-6.1.2/bin/bgp
               Instance #: 3
               Version ID: 00.00.0000
                  Respawn: ON
            Respawn count: 1
             Last started: Thu Jan  1 13:19:01 2017
            Process state: Run
            Package state: Normal
       Started on config: IPv6
            Feature name: ON
                     Tag : IPv6
            Process group: Group_10_bgp2
                     core: MAINMEM
                Max. core: 0
                Placement: Placeable
             startup_path: /pkg/startup/bgp.startup
                    Ready: 0.339s
                Available: 27.518s
        Process cpu time: 0.170 user, 0.080 kernel, 0.250 total
JID    TID  Stack pri state         TimeInState       HR:MM:SS:MSEC NAME
1054   1     328K  10 Receive         0:00:00:0099    0:00:00:0200 bgp
```

You can also check list of currently running BGP instances:

```
RP/0/0/CPU0:SP1R1#show bgp instances
[…]
Number of BGP instances: 2

ID  Placed-Grp  Name              AS        VRFs    Address Families
-------------------------------------------------------------------------------
0   v4_routing  IPv4              3356      0       IPv4 Unicast
1   bgp2_1      IPv6              3356      0       IPv6 Unicast
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

IOS XR has also ability to run different ASNs per instance, but again, some limitations apply. The most important one is that as IPv4 unicast and IPv6 unicast AFs can't be shared among more than one instance, the other instances are useable only for link state and VPNv4/v6 instances.

Here's example where there are multiple configured ASNs:

```
RP/0/0/CPU0:SP1R1#sh bgp instances

Number of BGP instances: 4

ID   Placed-Grp  Name              AS        VRFs    Address Families
-------------------------------------------------------------------------------
0    v4_routing  IPv4              3356      0       IPv4 Unicast
1    bgp2_1      IPv6              3356      0       IPv6 Unicast
2    bgp3_1      TEST              7711      0       VPNv6 Unicast
3    bgp4_1      default           179       0       Link-state Link-state
```

Please delete all instances and get back to one single instance running IPv4 and IPv6 before continuing to next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.5: BGP multi-session capability

In IOS-XE there's no option to scale the BGP out using separate BGP instances. We can however split sessions between pair of neighbors for any running address families. The feature is called multi-session.

To be able to run it, you need to activate at least two different address families for a neighbor. Let's check dummy configuration for four different AFs between CER7 and CER6:

```
CEL7#conf t
CEL7(config)#router bgp 100
CEL7(config-router)#address-family ipv6
CEL7(config-router-af)#neighbor 172.16.6.6 activate
CEL7(config-router-af)#address-family vpnv4
CEL7(config-router-af)#neighbor 172.16.6.6 activate
CEL7(config-router-af)#address-family vpnv6
CEL7(config-router-af)#neighbor 172.16.6.6 activate
CEL6#conf t
CEL6(config)#router bgp 100
CEL6(config-router)#address-family ipv6
CEL6(config-router-af)#neighbor 172.16.7.7 activate
CEL6(config-router-af)#address-family vpnv4
CEL6(config-router-af)#neighbor 172.16.7.7 activate
CEL6(config-router-af)#address-family vpnv6
CEL6(config-router-af)#neighbor 172.16.7.7 activate
```

When multi session capability is not enabled, you should see something like this:

```
CEL7#show ip bgp neighbors 172.16.6.6 | include [Ss]ession|address family
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
 For address family: IPv4 Unicast
  Session: 172.16.6.6
 For address family: IPv6 Unicast
  Session: 172.16.6.6
 For address family: VPNv4 Unicast
  Session: 172.16.6.6
 For address family: VPNv6 Unicast
  Session: 172.16.6.6
  Last reset 00:02:32, due to Capability changed of session 1
```

To enable multi session, you need to enable on both sides of configuration:

```
CEL6#conf t
CEL6(config)#router bgp 100
CEL6(config-router)#neighbor 172.16.7.7 transport multi-session
CEL7#conf t
CEL7(config)#router bgp 100
CEL7(config-router)#neighbor 172.16.6.6 transport multi-session
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Now, the output of previous command should change:

```
CEL7#show ip bgp neighbors 172.16.6.6 | include [Ss]ession|address family
  Session state = Established, up for 00:01:09
  BGP multisession with 4 sessions (4 established), first up for 00:01:09
  Neighbor sessions:
    4 active, is multisession capable
    Route refresh: advertised and received(new) on session 1, 2, 3, 4
    Four-octets ASN Capability: advertised and received on session 1, 2, 3, 4
    Multisession Capability: advertised and received
    Stateful switchover support enabled: NO for session 1, 2, 3, 4
  Message statistics for 172.16.6.6 session 1, state Established:
  Message statistics for 172.16.6.6 session 2, state Established:
  Message statistics for 172.16.6.6 session 3, state Established:
  Message statistics for 172.16.6.6 session 4, state Established:
 For address family: IPv4 Unicast
  Session: 172.16.6.6 session 1
  session 1 member
 For address family: IPv6 Unicast
  Session: 172.16.6.6 session 2
  session 2 member
 For address family: VPNv4 Unicast
  Session: 172.16.6.6 session 3
  session 3 member
 For address family: VPNv6 Unicast
  Session: 172.16.6.6 session 4
  session 4 member
  Last reset 00:01:55, due to Capability changed of session 1
```

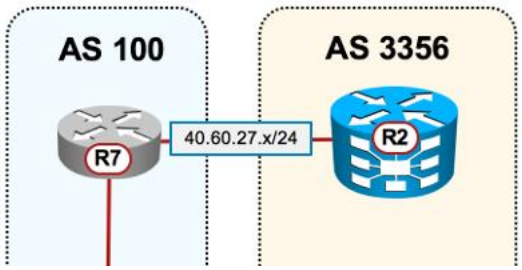

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.6: ORF: Optimal Route Filtering

While BGP peers can use inbound prefix and attribute filtering, this may create additional, serious load in situations where the advertisements contain large number of routes and at the same time, platform on the receiving end has way lower route processor capabilities.

To overcome such scenarios, characteristic for CE-PE deployments, we can use ORF – additional capability in BGP that can send defined filter (prefix-list) to neighbor and ask him to do the filtering for us.

Let's focus on AS100<>AS3356 peering with CEL7 and SP1R2:



SP1R2 sends following prefixes to CEL7:

```
RP/0/0/CPU0:SP1R2#sh bgp ipv4 unicast neighbors 40.60.27.7 advertised-routes
Network            Next Hop        From            AS Path
12.39.0.0/16       40.60.27.2      33.56.5.5       3356 1239i
33.56.0.0/16       40.60.27.2      Local Aggregate 3356i
44.44.67.0/24      40.60.27.2      41.61.26.6      3356 100i
66.66.67.0/24      40.60.27.2      33.56.5.5       3356 1239 200i
172.16.6.6/32      40.60.27.2      41.61.26.6      3356 100i
172.16.7.7/32      40.60.27.2      41.61.26.6      3356 100i
192.168.6.6/32     40.60.27.2      33.56.5.5       3356 1239 200i
192.168.7.7/32     40.60.27.2      33.56.5.5       3356 1239 200i

Processed 8 prefixes, 8 paths
```

On CEL7 we see only 5 prefixes and that's because 44.44.67.0/24, 172.16.6.6/32 and 172.16.7.7/32 are AS100 own prefixes and CEL7 detects an AS loop in AS_PATH:

```
CEL7#sh bgp ipv4 unicast neighbors 40.60.27.2 routes
[…]

     Network          Next Hop            Metric LocPrf Weight Path
 *>   12.39.0.0/16     40.60.27.2                             0 3356 1239 i
 *>   33.56.0.0/16     40.60.27.2                             0 3356 i
 *>   66.66.67.0/24    40.60.27.2                             0 3356 1239 200 i
 *>   192.168.6.6/32   40.60.27.2                             0 3356 1239 200 i
 *>   192.168.7.7/32   40.60.27.2                             0 3356 1239 200 i

Total number of prefixes 5
```

While 5 prefixes are not too much, imagine now that SP1R2 would send 5000 prefixes out of which CEL7 would only find a use of 5 or 10. What a waste of CPU processing power, bandwidth and convergence time!

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

We can of course filter on CEL7 side – let's assume we're interested only in the 33.56/16 and 12.39/16 prefixes. So, defining filter on CEL7:

```
CEL7(config)#ip prefix-list BGP-UPSTREAM permit 12.39.0.0/16 le 32
CEL7(config)#ip prefix-list BGP-UPSTREAM permit 33.56.0.0/16 le 32
CEL7(config)#router bgp 100
CEL7(config-router)#address-family ipv4 unicast
CEL7(config-router-af)#neighbor 40.60.27.2 prefix-list BGP-UPSTREAM in
```

Now, let's clear the session information inbound using BGP soft reconfig (utilizing BGP Route Refresh capability):

```
CEL7#clear ip bgp 40.60.27.2 soft in
```

We can see 2 prefixes:

```
CEL7#sh bgp ipv4 unicast neighbors 40.60.27.2 routes
BGP table version is 28, local router ID is 172.16.7.7
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>   12.39.0.0/16     40.60.27.2                             0 3356 1239 i
 *>   33.56.0.0/16     40.60.27.2                             0 3356 i

Total number of prefixes 2
```

However, the rest was denied by CEL7, so sent anyway from SP1R2 towards CEL7:

```
CEL7#sh ip bgp neighbors 40.60.27.2
BGP neighbor is 40.60.27.2,  remote AS 3356, external link
  BGP version 4, remote router ID 33.56.2.2
  BGP state = Established, up for 1d15h
[…]
                                 Outbound    Inbound
  Local Policy Denied Prefixes:  --------    -------
    prefix-list                         0          3
    AS_PATH loop:                     n/a         35
    Bestpath from this peer:           15        n/a
    Total:                             15         38
  Number of NLRIs in the update sent: max 3, min 0
```

The same effect can be however achieved by sending the prefix-list to SP1R2 from CEL7 utilizing ORF capability:

```
CEL7(config)#router bgp 100
CEL7(config-router)#address-family ipv4 unicast
CEL7(config-router-af)#neighbor 40.60.27.2 capability orf prefix-list ?
  both     Capability to SEND and RECEIVE the ORF to/from this neighbor
  receive  Capability to RECEIVE the ORF from this neighbor
  send     Capability to SEND the ORF to this neighbor
CEL7(config-router-af)#neighbor 40.60.27.2 capability orf prefix-list both
```

However, the ORF needs to be advertised also by the other side before it will be effective. Right now, CEL7 reports that ORF capability is only advertised to the neighbor and not received from the neighbor:

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CEL7#sh ip bgp neighbors
BGP neighbor is 40.60.27.2,  remote AS 3356, external link
  BGP version 4, remote router ID 33.56.2.2
  BGP state = Established, up for 00:00:27
[…]

 For address family: IPv4 Unicast
[…]
  10 update-group member
  AF-dependant capabilities:
    Outbound Route Filter (ORF) type (128) Prefix-list:
      Send-mode: advertised
      Receive-mode: advertised
```

…with the end effect of CEL7 needing still to filter out the prefixes:

```
CEL7#sh ip bgp neighbors | i Inb|Denie|prefix
  Incoming update prefix filter list is BGP-UPSTREAM
                                    Outbound    Inbound
  Local Policy Denied Prefixes:     --------    -------
    prefix-list                            0          3
```

Let's configure ORF capability on SP1R2:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#capability orf prefix both
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#
```



Note that in addition to normal neighbor route-policy, with ORF you can additionally add route-policy to filter additional routes. To do that, use following syntax:

```
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#orf route-policy ORF-POLICY
```

We should have ORF negotiated for neighbor and receive filter-list from CEL7:

```
RP/0/0/CPU0:SP1R2#sh bgp ipv4 unicast neighbors 40.60.27.7 | utility egrep "ORF|-mode"
Outbound Route Filter (ORF) type (128) Prefix:
      Send-mode: advertised, received
      Receive-mode: advertised, received
  Outbound Route Filter (ORF): received (2 entries)
```

And the filtering is successful, as we're advertising only two requested networks from SP1R2 towards CEL7:

```
RP/0/0/CPU0:SP1R2#sh bgp ipv4 unicast neighbors 40.60.27.7 advertised-routes
Network            Next Hop        From            AS Path
12.39.0.0/16       40.60.27.2      33.56.5.5       3356 1239i
33.56.0.0/16       40.60.27.2      Local Aggregate 3356i

Processed 2 prefixes, 2 paths
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

And on CEL7:

```
CEL7#sh bgp ipv4 unicast neighbors 40.60.27.2 routes
[…]

     Network          Next Hop            Metric LocPrf Weight Path
 *>   12.39.0.0/16     40.60.27.2                           0 3356 1239 i
 *>   33.56.0.0/16     40.60.27.2                           0 3356 i

Total number of prefixes 2
```

Please roll back Task changes before moving to the next task

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 7: MPLS Basics

This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch07.virl1.virl`
- **For VIRL²/CML²:** `sp.bootcamp.ch07.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch07.unl`

Task 1: Basic LDP configuration for label distribution

Our goal is to enable LDP protocol on interfaces within AS100, AS3356, AS1239 and AS200.

Task 1.1: Manually configure LDP protocol on IOS-XE

On IOS-XE, the easiest way to configure MPLS LDP is to enable MPLS IP on interfaces. LDP will be autoselected as label protocol and started on the interfaces:

```
CEL7(config)#interface gi3
CEL7(config-if)#mpls ip
```

And on CEL6:

```
CEL6(config)#interface gi3
CEL6(config-if)#mpls ip
```

Number of things happens now – LDP is assigned to run on the interface, discovers its neighbors and establishes session. Let's check what is going on.

First, the interfaces:

```
CEL6#sh mpls interfaces
Interface              IP            Tunnel   BGP Static Operational
GigabitEthernet3       Yes (ldp)     No       No  No     Yes
```

We see that GigabitEthernet 3 is running MPLS and the protocol is LDP. Then, let's check LDP state on the interfaces it's enabled:

```
CEL6#sh mpls ldp discovery
 Local LDP Identifier:
    172.16.6.6:0
    Discovery Sources:
    Interfaces:
        GigabitEthernet3 (ldp): xmit/recv
            LDP Id: 172.16.7.7:0
```

We see that over GigabitEthernet 3 we have one neighbor – 172.16.7.7 (as LDP by default uses router-id from Loopback interfaces). Let's check the neighbor:

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CEL6#sh mpls ldp neighbor
    Peer LDP Ident: 172.16.7.7:0; Local LDP Ident 172.16.6.6:0
        TCP connection: 172.16.7.7.33343 - 172.16.6.6.646
        State: Oper; Msgs sent/rcvd: 14/14; Downstream
        Up time: 00:05:58
        LDP discovery sources:
          GigabitEthernet3, Src IP addr: 44.44.67.7
        Addresses bound to peer LDP Ident:
          40.60.27.7      44.44.67.7      22.42.37.7      172.16.7.7
```

The connection is between 172.16.7.7 33343/tcp and 172.16.6.6 646/tcp. This is very important information with regards to any filtering on the interfaces – and may be the case in both real and lab network.

Let's check which labels were assigned by LDP to specific prefixes:

```
CEL6#sh mpls ldp bindings
  lib entry: 21.41.46.0/24, rev 2
        local binding:  label: imp-null
  lib entry: 22.42.37.0/24, rev 11
        no local binding
        remote binding: lsr: 172.16.7.7:0, label: imp-null
  lib entry: 40.60.27.0/24, rev 12
        no local binding
        remote binding: lsr: 172.16.7.7:0, label: imp-null
  lib entry: 41.61.16.0/24, rev 4
        local binding:  label: imp-null
  lib entry: 44.44.67.0/24, rev 6
        local binding:  label: imp-null
        remote binding: lsr: 172.16.7.7:0, label: imp-null
  lib entry: 172.16.6.6/32, rev 8
        local binding:  label: imp-null
        remote binding: lsr: 172.16.7.7:0, label: 16
  lib entry: 172.16.7.7/32, rev 10
        local binding:  label: 16
        remote binding: lsr: 172.16.7.7:0, label: imp-null
```



Cisco devices allocate labels for every prefix advertised by IGP protocol. Juniper devices allocate labels for every loopback interface, and then for all prefixes that have label assigned by the downstream router.

This difference has historical background. Cisco devices were first to forward MPLS traffic still using software implementation, where number of possible labels was not an issue (bound only by RAM memory).

Juniper devices, right from M40, were based on hardware forwarding and the number of labels was limited. JunOS default option was then to conserve the label space as much as it's possible without sacrificing features.

Please note, that thanks to LDP neighbor communication, we communicated our label binding of 16 to the 172.16.7.7/32 prefix and our peer communicated us his binding for 172.16.6.6/32 (our loopback) – with value of 16.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

We can check the MPLS forwarding information based on the LDP information we've seen:

```
CEL6#sh mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         Pop Label  172.16.7.7/32    0             Gi3        44.44.67.7
```

Additionally, to allocate labels only for /32 (and /128 for IPv6) prefixes, you need to change the default policy:

```
SP1-R5-RR(config)#mpls ldp label
SP1-R5-RR(config-ldp-lbl)#allocate global host-routes
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: Manually configure LDP protocol on IOS XR

On IOS XR the configuration is feature-centric, in line with the IOS XR concepts. So, we need to enter MPLS LDP configuration and then assign interfaces to LDP:

```
RP/0/0/CPU0:SP1R1(config)#mpls ldp
RP/0/0/CPU0:SP1R1(config-ldp)#interface gigabitEthernet 0/0/0/0
RP/0/0/CPU0:SP1R1(config-ldp-if)#interface gigabitEthernet 0/0/0/2
RP/0/0/CPU0:SP1R1(config-ldp-if)#interface gigabitEthernet 0/0/0/3
RP/0/0/CPU0:SP1R1(config-ldp-if)#interface gigabitEthernet 0/0/0/5
RP/0/0/CPU0:SP1R1(config-ldp-if)#
```

We can observe similar information as with IOS-XE. First – the information reported by LDP discovery:

```
RP/0/0/CPU0:SP1R1#show mpls ldp discovery

Local LDP Identifier: 33.56.1.1:0
Discovery Sources:
  Interfaces:
    GigabitEthernet0/0/0/0 : xmit/recv
      VRF: 'default' (0x60000000)
      LDP Id: 33.56.4.4:0, Transport address: 33.56.4.4
          Hold time: 15 sec (local:15 sec, peer:15 sec)
          Established: Jan  1 20:51:39.236 (00:02:33 ago)

    GigabitEthernet0/0/0/2 : xmit/recv
      VRF: 'default' (0x60000000)
      LDP Id: 33.56.2.2:0, Transport address: 33.56.2.2
          Hold time: 15 sec (local:15 sec, peer:15 sec)
          Established: Jan  1 20:50:55.939 (00:03:16 ago)

    GigabitEthernet0/0/0/3 : xmit/recv
      VRF: 'default' (0x60000000)
      LDP Id: 33.56.5.5:0, Transport address: 33.56.5.5
          Hold time: 15 sec (local:15 sec, peer:15 sec)
          Established: Jan  1 20:51:49.085 (00:02:23 ago)

    GigabitEthernet0/0/0/5 : xmit/recv
      VRF: 'default' (0x60000000)
      LDP Id: 33.56.3.3:0, Transport address: 33.56.3.3
          Hold time: 15 sec (local:15 sec, peer:15 sec)
          Established: Jan  1 20:51:18.587 (00:02:53 ago)
```

Next, LDP neighbor information:

```
RP/0/0/CPU0:SP1R1#sh mpls ldp neighbor

Peer LDP Identifier: 33.56.2.2:0
  TCP connection: 33.56.2.2:33254 - 33.56.1.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 22/22; Downstream-Unsolicited
  Up time: 00:05:00
  LDP Discovery Sources:
    IPv4: (1)
      GigabitEthernet0/0/0/2
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (5)
      33.56.2.2       33.56.12.2      33.56.23.2      33.56.24.2
      40.60.27.2
    IPv6: (0)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Peer LDP Identifier: 33.56.3.3:0
  TCP connection: 33.56.3.3:35899 - 33.56.1.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 22/24; Downstream-Unsolicited
  Up time: 00:04:38
  LDP Discovery Sources:
    IPv4: (1)
      GigabitEthernet0/0/0/5
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (8)
      10.0.128.1     22.22.33.1     22.42.37.3     33.56.3.3
      33.56.13.3     33.56.23.3     33.56.34.3     33.56.35.3
    IPv6: (0)

[…]
```

And now for the bindings created by LDP:

```
RP/0/0/CPU0:SP1R1#show mpls ldp bindings brief

Prefix              Local      Advertised  Remote Bindings
                    Label        (peers)       (peers)
------------------  ---------  ----------  ---------------
10.0.128.0/29       -                   0                2
11.11.11.0/24       -                   0                1
22.22.33.0/24       -                   0                1
22.42.37.0/24       -                   0                1
33.56.1.1/32        ImpNull             4                4
33.56.2.2/32        24001               4                4
33.56.3.3/32        24002               4                4
33.56.4.4/32        24005               4                4
33.56.5.5/32        24000               4                4
33.56.12.0/24       ImpNull             4                4
33.56.13.0/24       ImpNull             4                4
33.56.14.0/24       ImpNull             4                4
33.56.15.0/24       ImpNull             4                4
33.56.23.0/24       24003               4                4
33.56.24.0/24       24006               4                4
33.56.34.0/24       24007               4                4
33.56.35.0/24       24004               4                4
33.56.44.44/32      -                   0                1
40.60.27.0/24       -                   0                1
41.61.16.0/24       ImpNull             4                0
41.61.46.0/24       -                   0                1
```

As you can see, IOS XR will by default assign labels for any prefix. To allocate labels only for /32 prefixes (host routes – we'll get back to that for IPv6) execute:

```
RP/0/0/CPU0:SP1R4(config)#mpls ldp
RP/0/0/CPU0:SP1R4(config-ldp)#address-family ipv4
RP/0/0/CPU0:SP1R4(config-ldp-af)#label local allocate for host-routes
```

Please repeat this for other AS3356 routers.

Another useful command for LDP is '**brief**' version of MPLS output – it gives nice summary of number of labels allocated per each neighbor:

```
RP/0/0/CPU0:SP1R1#sh mpls ldp neighbor brief

Peer                GR   NSR  Up Time     Discovery   Addresses     Labels
                                           ipv4  ipv6  ipv4  ipv6  ipv4   ipv6
------------------  ---  ---  ----------  ----------  ----------  ------------
33.56.2.2:0         N    N    00:00:09    1     0     5     0     5      0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
33.56.4.4:0          N    N     00:00:08    1      0      8      0      6      0
33.56.5.5:0          N    N     00:00:08    1      0      3      0      5      0
33.56.3.3:0          N    N     00:00:08    1      0      8      0      5      0
```

When LDP refreshes, the list of allocated labels will be shorter:

```
RP/0/0/CPU0:SP1R1#sh mpls ldp bindings brief

Prefix              Local      Advertised  Remote Bindings
                    Label        (peers)       (peers)
------------------  ---------  ----------  ---------------
33.56.1.1/32        ImpNull             4                4
33.56.2.2/32        24001               4                4
33.56.3.3/32        24002               4                4
33.56.4.4/32        24005               4                4
33.56.5.5/32        24000               4                4
33.56.44.44/32      -                   0                1
```

Small hint, as we're starting to work with MPLS - you may hit some problems with MPLS labels creating additional stack. If you're troubleshooting this problem and suspect one of the MTU settings have been changed, you can verify this by asking XR interface database:

```
RP/0/0/CPU0:SP1R2#sh im database interface gigabitEthernet 0/0/0/0

View: OWN - Owner, L3P - Local 3rd Party, G3P - Global 3rd Party, LDP - Local Data Plane
      GDP - Global Data Plane, RED - Redundancy, UL - UL

Node 0/0/CPU0 (0x0)

Interface GigabitEthernet0/0/0/0, ifh 0x00000040 (up, 1514)
  Interface flags:          0x000000000150059f (ROOT_IS_HW|PHYS_ON_RP
                            |IFCONNECTOR|IFINDEX|SUP_NAMED_SUB|BROADCAST
                            |CONFIG|HW|VIS|DATA|CONTROL)
  Encapsulation:            ether
  Interface type:           IFT_GETHERNET
  Control parent:           None
  Data parent:              None
  Views:                    UL|GDP|LDP|G3P|L3P|OWN

  Protocol        Caps (state, mtu)
  --------        -----------------
  None            spio (up, 1514)
  None            ether (up, 1514)
  arp             arp (up, 1500)
  ipv4            ipv4 (up, 1500)
  mpls            mpls (up, 1500)
  ipv6            ipv6_preswitch (up, 1500)
  ipv6            ipv6 (up, 1500)
  ether_sock      ether_sock (up, 1500)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3: LDP RID vs transport address

By default, LDP both chooses RID based on Loopback0 interface and uses the address to establish the TCP connection.

Sometimes, RID is not carried over on specific link and routers may be simply missing routing information for that address. You can therefore choose to use other addressing as transport address information for LDP session.

On CEL6 we have default configuration – RID is used both as LDP ID and as the address of transport connection endpoint:

```
CEL6#sh mpls ldp discovery
 Local LDP Identifier:
    172.16.6.6:0                        ! R6 loopback0 address
    Discovery Sources:
    Interfaces:
        GigabitEthernet2 (ldp): xmit
        GigabitEthernet3 (ldp): xmit/recv
            LDP Id: 172.16.7.7:0        ! R7 loopback0 address
```

This is matched with CEL7 router LDP information:

```
CEL7#sh mpls ldp discovery
 Local LDP Identifier:
    172.16.7.7:0                        ! R7 loopback0 address
    Discovery Sources:
    Interfaces:
        GigabitEthernet3 (ldp): xmit/recv
            LDP Id: 172.16.6.6:0        ! R6 loopback0 address
```

To change the address transport layer uses to establish a session, you configure this on the node and interface you want to change. In this example, we'll force the session to use CEL7 Gi3 interface IP address to establish a session to CEL6:

```
CEL7(config)#int gi3
CEL7(config-if)#mpls ldp discovery transport-address interface
```

The session should go down and then up in a moment:

```
*Jan 1 15:27:08.020: %LDP-5-NBRCHG: LDP Neighbor 172.16.6.6:0 (1) is DOWN (Received
error notification from peer: Holddown time expired)
*Jan 1 15:27:11.334: %LDP-5-NBRCHG: LDP Neighbor 172.16.6.6:0 (2) is UP
```

Notice the RID didn't change. Also, from CEL7 perspective, the addressing LDP uses still uses RID address:

```
CEL7#sh mpls ldp discovery
 Local LDP Identifier:
    172.16.7.7:0
    Discovery Sources:
    Interfaces:
        GigabitEthernet3 (ldp): xmit/recv
            LDP Id: 172.16.6.6:0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

However, on the CEL6 the view did change:

```
CEL6#sh mpls ldp discovery
 Local LDP Identifier:
    172.16.6.6:0
    Discovery Sources:
    Interfaces:
        GigabitEthernet2 (ldp): xmit
        GigabitEthernet3 (ldp): xmit/recv
            LDP Id: 172.16.7.7:0; IP addr: 44.44.67.7
```

The same can be observed on IOS XR. Let's take a look at the situation between SP1R1 and SP1R5:

```
RP/0/0/CPU0:SP1R1#sh mpls ldp discovery gigabitEthernet 0/0/0/3
Local LDP Identifier: 33.56.1.1:0
Discovery Source:
    GigabitEthernet0/0/0/3 : xmit/recv
      VRF: 'default' (0x60000000)
      LDP Id: 33.56.5.5:0, Transport address: 33.56.5.5
          Hold time: 15 sec (local:15 sec, peer:15 sec)
          Established: Jan 14 10:10:58.322 (07:03:12 ago)

SP1-R5-RR#sh mpls ldp discovery detail
 Local LDP Identifier:
    33.56.5.5:0
    Discovery Sources:
    Interfaces:
        GigabitEthernet2 (ldp): xmit/recv
            Enabled: IGP config;
            Hello interval: 5000 ms; Transport IP addr: 33.56.5.5
            LDP Id: 33.56.3.3:0
              Src IP addr: 33.56.35.3; Transport IP addr: 33.56.3.3
              Hold time: 15 sec; Proposed local/peer: 15/15 sec
              Reachable via 33.56.3.3/32
              Password: not required, none, in use
            Clients: IPv4, mLDP
        GigabitEthernet3 (ldp): xmit/recv
            Enabled: IGP config;
            Hello interval: 5000 ms; Transport IP addr: 33.56.5.5
            LDP Id: 33.56.1.1:0
              Src IP addr: 33.56.15.1; Transport IP addr: 33.56.1.1
              Hold time: 15 sec; Proposed local/peer: 15/15 sec
              Reachable via 33.56.1.1/32
              Password: not required, none, in use
            Clients: IPv4, mLDP
```

Changing on IOS XR side:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#mpls ldp
RP/0/0/CPU0:SP1R1(config-ldp)#interface gigabitEthernet 0/0/0/3
RP/0/0/CPU0:SP1R1(config-ldp-if)#address-family ipv4
RP/0/0/CPU0:SP1R1(config-ldp-if-af)#discovery transport-address interface
RP/0/0/CPU0:SP1R1(config-ldp-if-af)#commit
RP/0/0/CPU0:SP1R1#clear mpls ldp neighbor 33.56.5.5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Now, let's verify the change:

```
RP/0/0/CPU0:SP1R1#sh mpls ldp discovery gigabitEthernet 0/0/0/3 detail
Wed Jan 14 17:20:03.268 UTC

Local LDP Identifier: 33.56.1.1:0
Discovery Source:
    GigabitEthernet0/0/0/3 (0xa0) : xmit/recv
      VRF: 'default' (0x60000000)
      Source address: 33.56.15.1; Transport address: 33.56.15.1
      Hello interval: 5 sec (due in 4 sec)
      Quick-start: Enabled
      LDP Id: 33.56.5.5:0
          Source address: 33.56.15.5; Transport address: 33.56.5.5
          Hold time: 15 sec (local:15 sec, peer:15 sec)
                     (expiring in 13.8 sec)
          Established: Jan 14 17:16:29.203 (00:03:34 ago)
          Last session connection failures:
             Jan  1 07:11:06.444: User cleared session manually
                 (Last up for 00:01:08)

SP1-R5-RR#sh mpls ldp discovery detail
 Local LDP Identifier:
    33.56.5.5:0
    Discovery Sources:
    Interfaces:
        GigabitEthernet2 (ldp): xmit/recv
            Enabled: IGP config;
            Hello interval: 5000 ms; Transport IP addr: 33.56.5.5
            LDP Id: 33.56.3.3:0
              Src IP addr: 33.56.35.3; Transport IP addr: 33.56.3.3
              Hold time: 15 sec; Proposed local/peer: 15/15 sec
              Reachable via 33.56.3.3/32
              Password: not required, none, in use
            Clients: IPv4, mLDP
        GigabitEthernet3 (ldp): xmit/recv
            Enabled: IGP config;
            Hello interval: 5000 ms; Transport IP addr: 33.56.5.5
            LDP Id: 33.56.1.1:0
              Src IP addr: 33.56.15.1; Transport IP addr: 33.56.15.1
              Hold time: 15 sec; Proposed local/peer: 15/15 sec
              Reachable via 33.56.15.0/24
              Password: not required, none, in use
            Clients: IPv4, mLDP
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4: Using MPLS LDP autoconfig on IOS-XE

Another option for configuring the LDP is to use autoconfiguration. In this mode, LDP is automatically started on all interfaces covered by specified IGP protocol.

Let's do this by deleting the MPLS configuration from GigabitEthernet2 and GigabitEthernet3 interfaces on SP1R5:

```
SP1-R5-RR(config)#int gi2
SP1-R5-RR(config-if)#no mpls ip
*Jan  1 22:58:20.622: %LDP-5-NBRCHG: LDP Neighbor 33.56.3.3:0 (1) is DOWN (LDP disabled
on interface)
SP1-R5-RR(config-if)#int gi3
SP1-R5-RR(config-if)#no mpls ip
*Jan  1 22:58:23.310: %LDP-5-NBRCHG: LDP Neighbor 33.56.1.1:0 (2) is DOWN (LDP disabled
on interface)
```

Both LDP sessions went down.

Now, let's enable LDP autoconfiguration under OSPFv2 process:

```
SP1-R5-RR(config)#router ospf 100
SP1-R5-RR(config-router)#mpls ldp autoconfig
SP1-R5-RR(config-router)#^Z
*Jan  1 22:58:44.869: %LDP-5-NBRCHG: LDP Neighbor 33.56.3.3:0 (1) is UP
*Jan  1 22:58:44.877: %LDP-5-NBRCHG: LDP Neighbor 33.56.1.1:0 (2) is UP
SP1-R5-RR#
```

LDP sessions are established again:

```
SP1-R5-RR#sh mpls ldp neighbor
    Peer LDP Ident: 33.56.3.3:0; Local LDP Ident 33.56.5.5:0
        TCP connection: 33.56.3.3.646 - 33.56.5.5.34789
        State: Oper; Msgs sent/rcvd: 13/13; Downstream
        Up time: 00:05:00
        LDP discovery sources:
          GigabitEthernet2, Src IP addr: 33.56.35.3
        Addresses bound to peer LDP Ident:
          33.56.3.3       33.56.23.3      22.22.33.1      33.56.34.3
          33.56.35.3      10.0.128.1      22.42.37.3      33.56.13.3
    Peer LDP Ident: 33.56.1.1:0; Local LDP Ident 33.56.5.5:0
        TCP connection: 33.56.1.1.646 - 33.56.5.5.43281
        State: Oper; Msgs sent/rcvd: 13/13; Downstream
        Up time: 00:05:00
        LDP discovery sources:
          GigabitEthernet3, Src IP addr: 33.56.15.1
        Addresses bound to peer LDP Ident:
          33.56.1.1       33.56.14.1      41.61.16.1      33.56.12.1
          33.56.15.1      33.56.13.1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.5: Using MPLS LDP autoconfig on IOS XR

As with IOS-XE, IOS XR can also use LDP autoconfiguration. The configuration is similar – instead of configuring LDP interfaces statically, you use IGP routing configuration section. For SP1R1 the configuration will be:

```
RP/0/0/CPU0:SP1R1(config)#mpls ldp
RP/0/0/CPU0:SP1R1(config-ldp)#no int gi0/0/0/0
RP/0/0/CPU0:SP1R1(config-ldp)#no int gi0/0/0/2
RP/0/0/CPU0:SP1R1(config-ldp)#no int gi0/0/0/3
RP/0/0/CPU0:SP1R1(config-ldp)#no int gi0/0/0/5
RP/0/0/CPU0:SP1R1(config-ldp)#exit
RP/0/0/CPU0:SP1R1(config)#router ospf AS3356
RP/0/0/CPU0:SP1R1(config-ospf)#mpls ldp auto-config
RP/0/0/CPU0:SP1R1(config-ospf)#commit
```

The result will be similar to our static config but will work dynamically in line with IGP changes. Notice that the list of LDP enabled interfaces is exactly the same as with static configuration:

```
RP/0/0/CPU0:SP1R1#sh mpls ldp interface brief
Interface       VRF Name             Config Enabled IGP-Auto-Cfg TE-Mesh-Grp cfg
--------------- -------------------- ------ ------- ------------ ---------------
Gi0/0/0/0       default              N      Y       1            N/A
Gi0/0/0/1       default              N      N       0            N/A
Gi0/0/0/2       default              N      Y       1            N/A
Gi0/0/0/3       default              N      Y       1            N/A
Gi0/0/0/4       default              N      N       0            N/A
Gi0/0/0/5       default              N      Y       1            N/A
```

Please reconfigure all IOS XR routers in AS3356 to use MPLS LDP autoconfiguration.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.6: Control label range for LDP

Range of assigned labels can be changed with following command on IOS-XE:

```
SP1-R5-RR(config)#mpls label range 5500 5599
```

…and on IOS XR:

```
RP/0/0/CPU0:SP1R1(config)#mpls label range 101000 101999
```

Unfortunately, for this new allocation to take effect, router needs to be rebooted. Before doing so (and waiting for reboot), configure following ranges:

SP1R1 – 101000 to 101999

SP1R2 – 102000 to 102999

SP1R3 – 103000 to 103999

SP1R4 – 104000 to 104999

SP1R5 – 105000 to 105999

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.7: LDP sync

If there's a disconnect between LDP and IGP operating or not operating on a given link, MPLS network can encounter traffic drop due to missing or incorrect label information.

If the link is up from IGP perspective, and LDP is not yet fully converged, the LSR will strip labels and attempt IP forwarding for traffic that should be forwarded with labels – this will usually lead to traffic being dropped.

This may happen also because LDP configuration is out of sync on routers (for example – because of mismatched password) and IGP link exists (and is shortest path to destination) or some filtering is taking place.

MPLS LDP IGP sync will make sure that LDP fully converged over a link before it is used for forwarding – IGP needs to take care of advertising the link with maximum cost to avoid steering the traffic over it.

To configure LDP IGP sync you need to instruct IGP (OSPF or IS-IS) to turn the capability on. On IOS XR for OSPF it's done in this way:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#router ospf AS3356
RP/0/0/CPU0:SP1R1(config-ospf)#mpls ldp sync
```

When LDP IGP sync is disabled on IOS-XE, the LDP status from OSPF looks like this:

```
SP1-R5-RR#sh mpls ldp igp sync
    GigabitEthernet2:
        LDP configured; LDP-IGP Synchronization not enabled.
    GigabitEthernet3:
        LDP configured; LDP-IGP Synchronization not enabled.
```

When you enable the LDP IGP sync:

```
SP1-R5-RR(config)#router ospf 100
SP1-R5-RR(config-router)#mpls ldp sync
SP1-R5-RR(config-router)#^Z
```

The status changes to per-interface status of sync and reachability information:

```
SP1-R5-RR#sh mpls ldp igp sync
    GigabitEthernet2:
        LDP configured; LDP-IGP Synchronization enabled.
        Sync status: sync achieved; peer reachable.
        Sync delay time: 0 seconds (0 seconds left)
        IGP holddown time: infinite.
        Peer LDP Ident: 33.56.3.3:0
        IGP enabled: OSPF 100
    GigabitEthernet3:
        LDP configured; LDP-IGP Synchronization enabled.
        Sync status: sync achieved; peer reachable.
        Sync delay time: 0 seconds (0 seconds left)
        IGP holddown time: infinite.
        Peer LDP Ident: 33.56.1.1:0
        IGP enabled: OSPF 100
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.8: LDP session protection

When link between neighbor fails, LDP session will be at some point teared down also. If there's still a connection between nodes through some other paths, it may be beneficial to keep the label mappings (bindings) distributed previously using mechanism called LDP targeted hellos.

Targeted hellos are sessions set up using unicast between LDP nodes.

Let's configure LDP session protection in AS3356:

```
RP/0/0/CPU0:SP1R1(config)#mpls ldp
RP/0/0/CPU0:SP1R1(config-ldp)#session protection
RP/0/0/CPU0:SP1R2(config)#mpls ldp
RP/0/0/CPU0:SP1R2(config-ldp)#session protection
RP/0/0/CPU0:SP1R3(config)#mpls ldp
RP/0/0/CPU0:SP1R3(config-ldp)#session protection
RP/0/0/CPU0:SP1R4(config)#mpls ldp
RP/0/0/CPU0:SP1R4(config-ldp)#session protection

SP1-R5-RR(config)#mpls ldp session protection
```

The neighbor information will show additional, targeted session:

```
RP/0/0/CPU0:SP1R1#sh mpls ldp neighbor 33.56.4.4
Peer LDP Identifier: 33.56.4.4:0
  TCP connection: 33.56.4.4:28780 - 33.56.1.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 866/862; Downstream-Unsolicited
  Up time: 12:29:53
  LDP Discovery Sources:
    IPv4: (2)
      GigabitEthernet0/0/0/0
      Targeted Hello (33.56.1.1 -> 33.56.4.4, active)
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (8)
      10.0.128.2     11.11.11.1     33.56.4.4      33.56.14.4
      33.56.24.4     33.56.34.4     33.56.44.44    41.61.46.4
    IPv6: (0)
```

As soon as we'll shut down Gi3 on SP1R5 interface, the LDP reports it'll keep the targeted session up. And indeed, it keeps the session:

```
SP1-R5-RR(config)#int gi3
SP1-R5-RR(config-if)#shut
SP1-R5-RR(config-if)#^Z
*Jan  1 12:30:00.466: %LDP-5-SP: 33.56.1.1:0: session hold up initiated

SP1-R5-RR#sh mpls ldp neighbor
[…]
    Peer LDP Ident: 33.56.1.1:0; Local LDP Ident 33.56.5.5:0
        TCP connection: 33.56.1.1.646 - 33.56.5.5.40146
        State: Oper; Msgs sent/rcvd: 190/190; Downstream
        Up time: 02:40:07
        LDP discovery sources:
          Targeted Hello 33.56.5.5 -> 33.56.1.1, active, passive
        Addresses bound to peer LDP Ident:
          33.56.1.1       33.56.14.1      41.61.16.1      33.56.12.1
          33.56.15.1      33.56.13.1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

While in this state, LDP will retain label mappings. When we unshut the Gi3 interface on SP1R5, the LDP session recovery starts to establish direct adjacency:

```
SP1-R5-RR(config)#int gi3
SP1-R5-RR(config-if)#no shut
SP1-R5-RR(config-if)#^Z
*Jan  1 12:36:49.493: %LDP-5-SP: 33.56.1.1:0: session recovery succeeded
SP1-R5-RR#sh mpls ldp neighbor
[…]
    Peer LDP Ident: 33.56.1.1:0; Local LDP Ident 33.56.5.5:0
        TCP connection: 33.56.1.1.646 - 33.56.5.5.40146
        State: Oper; Msgs sent/rcvd: 199/198; Downstream
        Up time: 02:46:55
        LDP discovery sources:
          Targeted Hello 33.56.5.5 -> 33.56.1.1, active, passive
          GigabitEthernet3, Src IP addr: 33.56.15.1
        Addresses bound to peer LDP Ident:
          33.56.1.1       33.56.14.1      41.61.16.1      33.56.12.1
          33.56.15.1      33.56.13.1
```

The same can be observed for IOS XR based LDP session protection. Let's shut down link between SP1R1 and SP1R4 and observe the process.

Make sure we'll get debug output on console:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#logging console debugging
RP/0/0/CPU0:SP1R1(config)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
RP/0/0/CPU0:SP1R1(config)#
RP/0/0/CPU0:SP1R1#debug mpls ldp session protection
```

Let's check current status of LDP connection to 33.56.4.4:

```
RP/0/0/CPU0:SP1R1#sh mpls ldp neighbor 33.56.4.4
Peer LDP Identifier: 33.56.4.4:0
  TCP connection: 33.56.4.4:28780 - 33.56.1.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 894/890; Downstream-Unsolicited
  Up time: 12:54:09
  LDP Discovery Sources:
    IPv4: (2)
      GigabitEthernet0/0/0/0
      Targeted Hello (33.56.1.1 -> 33.56.4.4, active)
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (8)
      10.0.128.2      11.11.11.1      33.56.4.4       33.56.14.4
      33.56.24.4      33.56.34.4      33.56.44.44     41.61.46.4
    IPv6: (0)
```

When we shut down the link:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#int g0/0/0/0
RP/0/0/CPU0:SP1R1(config-if)#shut
RP/0/0/CPU0:SP1R1(config-if)#commit
```

We see from the debug (only relevant info shown):

```
RP/0/0/CPU0:Jan  1 12:43:31.978 : mpls_ldp[1181]: DBG-SP[1], Peer(33.56.4.4:0): last
primary adj lost; starting session protection holdup timer
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:Jan  1 12:43:31.978 : mpls_ldp[1181]: DBG-SP[1], Peer(33.56.4.4:0): State
change: 'Ready' -> 'Protecting'
```

The session is still protected:

```
RP/0/0/CPU0:SP1R1#sh mpls ldp neighbor 33.56.4.4
Peer LDP Identifier: 33.56.4.4:0
  TCP connection: 33.56.4.4:28780 - 33.56.1.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 896/890; Downstream-Unsolicited
  Up time: 12:54:42
  LDP Discovery Sources:
    IPv4: (1)
      Targeted Hello (33.56.1.1 -> 33.56.4.4, active)
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (8)
      10.0.128.2     11.11.11.1     33.56.4.4      33.56.14.4
      33.56.24.4     33.56.34.4     33.56.44.44    41.61.46.4
    IPv6: (0)
```

Bring the Gi0/0/0/0 interface back up on SP1R1 before proceeding.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.9: LDPv6 or labels for IPv6 prefixes

LDP was also standardized for IPv6 transport and exchanging labels for IPv6 prefixes. Note, that as of now it's only supported to IOS XR. To enable IPv6 features you need to enable IPv6 under MPLS LDP and activate IPv6 on the interfaces (we're assuming below you're doing manual configuration for IPv4 and IPv6):

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#mpls ldp
RP/0/0/CPU0:SP1R2(config-ldp)#address-family ipv6
RP/0/0/CPU0:SP1R2(config-ldp)#interface gigabitEthernet 0/0/0/0
RP/0/0/CPU0:SP1R2(config-ldp-if)#address-family ipv4
RP/0/0/CPU0:SP1R2(config-ldp-if-af)#address-family ipv6 ! repeat this configuration for
                                                        ! all core-facing interfaces and
                                                        ! other IOS XR routers
RP/0/0/CPU0:SP1R2(config-ldp-if-af)#commit
```

You can see the LDP IPv6 labels allocated locally by the router by executing explicit IPv6 command:

```
RP/0/0/CPU0:SP1R2#sh mpls ldp ipv6 forwarding
[…]

Prefix                Label   Label(s)       Outgoing     Next Hop            Flags
                      In      Out            Interface                        G S R
--------------------- ------- -------------- ------------ ------------------- -----
2001:33:56:13::/64    24017   ImpNull        Gi0/0/0/0    Link-local
                              ImpNull        Gi0/0/0/2    Link-local
2001:33:56:14::/64    24015   ImpNull        Gi0/0/0/2    Link-local
                              ImpNull        Gi0/0/0/4    Link-local
2001:33:56:15::/64    24016   ImpNull        Gi0/0/0/2    Link-local
2001:33:56:34::/64    24013   ImpNull        Gi0/0/0/0    Link-local
                              ImpNull        Gi0/0/0/4    Link-local
2001:33:56:35::/64    24014   ImpNull        Gi0/0/0/0    Link-local
2001:33:56:aa::1/128  24012   ImpNull        Gi0/0/0/2    Link-local
2001:33:56:aa::3/128  24010   ImpNull        Gi0/0/0/0    Link-local
2001:33:56:aa::4/128  24009   ImpNull        Gi0/0/0/4    Link-local
2001:33:56:aa::5/128  24011   24004          Gi0/0/0/0    Link-local
                              24004          Gi0/0/0/2    Link-local
```

Or more generally, using:

```
RP/0/0/CPU0:SP1R2#sh mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24000  Pop         33.56.4.4/32       Gi0/0/0/4    33.56.24.4      0
24001  Pop         33.56.3.3/32       Gi0/0/0/0    33.56.23.3      0
[…]
24009  Pop         2001:33:56:aa::4/128   \
                                      Gi0/0/0/4    fe80::4         948
24010  Pop         2001:33:56:aa::3/128   \
                                      Gi0/0/0/0    fe80::3         1026
24011  24004       2001:33:56:aa::5/128   \
                                      Gi0/0/0/0    fe80::3         1393
       24004       2001:33:56:aa::5/128   \
                                      Gi0/0/0/2    fe80::1         0
24012  Pop         2001:33:56:aa::1/128   \
                                      Gi0/0/0/2    fe80::1         1104
 […]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Note, that as of current version of the IOS XR used in practical exam, there's no support of IGP auto-config for MPLS LDP for IPv6.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Basic MPLS operations

Task 2.1: Checking MPLS label forwarding for LDP bindings

Let's to a path trace node by node and observe the Label Switched Path that LDP built for us. We'll start from SP1R2 towards SP1R5 – to have at least one hop to destination.

On SP1R2 we can see following LDP generated label mappings:

```
RP/0/0/CPU0:SP1R2#sh mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
102000 Exp-Null-v4 33.56.3.3/32       Gi0/0/0/0    33.56.23.3      219084
102001 Exp-Null-v4 33.56.1.1/32       Gi0/0/0/2    33.56.12.1      408772
102002 103000      33.56.5.5/32       Gi0/0/0/0    33.56.23.3      5607
       101000      33.56.5.5/32       Gi0/0/0/2    33.56.12.1      59998
102003 Exp-Null-v4 33.56.4.4/32       Gi0/0/0/4    33.56.24.4      217538
```

We see that router has one local label to 33.56.5.5/32 prefix – 102002. At the same time, if traffic would be generated, there will be two possible labels to choose – 103000 and 101000 advertised by SP1R3 (33.56.23.3) and SP1R1 (33.56.12.1) respectively.

To look at the same table but sorted by prefix, execute:

```
RP/0/0/CPU0:SP1R2#sh mpls ldp ipv4 forwarding

Codes:
  - = GR label recovering, (!) = LFA FRR pure backup path
  {} = Label stack with multi-line output for a routing path
  G = GR, S = Stale, R = Remote LFA FRR backup

Prefix           Label   Label(s)       Outgoing     Next Hop            Flags
                 In      Out            Interface                        G S R
---------------- ------- -------------- ------------ ------------------- -----
33.56.1.1/32     102001  ExpNullv4      Gi0/0/0/2    33.56.12.1
33.56.3.3/32     102000  ExpNullv4      Gi0/0/0/0    33.56.23.3
33.56.4.4/32     102003  ExpNullv4      Gi0/0/0/4    33.56.24.4
33.56.5.5/32     102002  103000         Gi0/0/0/0    33.56.23.3
                         101000         Gi0/0/0/2    33.56.12.1
```

To ask for mapping for specific, known local label do:

```
RP/0/0/CPU0:SP1R2#show mpls forwarding labels 102002
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
102002 103000      33.56.5.5/32       Gi0/0/0/0    33.56.23.3      5607
       101000      33.56.5.5/32       Gi0/0/0/2    33.56.12.1      61684
```

The 'detail' keyword added at the back of the command will show us also MTU and label stack data:

```
RP/0/0/CPU0:SP1R2#show mpls forwarding labels 102002 detail
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
102002 103000      33.56.5.5/32       Gi0/0/0/0    33.56.23.3      5607
     Updated: Jan  1 09:49:54.631
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
     Version: 48, Priority: 3
     Label Stack (Top -> Bottom): { 103000 }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 14/18, MTU: 1500
     Packets Switched: 84

       101000       33.56.5.5/32        Gi0/0/0/2     33.56.12.1       61839
     Updated: Jan  1 13:41:57.417
     Version: 48, Priority: 3
     Label Stack (Top -> Bottom): { 101000 }
     NHID: 0x0, Encap-ID: N/A, Path idx: 1, Backup path idx: 0, Weight: 0
     MAC/Encaps: 14/18, MTU: 1500
     Packets Switched: 804
```

We can also ask CEF to give us the label mappings per interface for specific address, like that:

```
RP/0/0/CPU0:SP1R2#sh cef ipv4 33.56.5.5
33.56.5.5/32, version 48, internal 0x1000001 0x0 (ptr 0xa143cef4) [1], 0x0 (0xa1407a94),
0xa28 (0xa18af050)
 Updated Jan  1 13:41:57.417
 local adjacency 33.56.23.3
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 33.56.23.3/32, GigabitEthernet0/0/0/0, 6 dependencies, weight 0, class 0 [flags
0x0]
    path-idx 0 NHID 0x0 [0xa10fb544 0x0]
    next hop 33.56.23.3/32
    local adjacency
     local label 102002      labels imposed {103000}
   via 33.56.12.1/32, GigabitEthernet0/0/0/2, 5 dependencies, weight 0, class 0 [flags
0x0]
    path-idx 1 NHID 0x0 [0xa10fb5ec 0x0]
    next hop 33.56.12.1/32
    local adjacency
     local label 102002      labels imposed {101000}
```

Let's follow the SP1R1 path – we know that SP1R1 will receive traffic labelled with 101000 (as SP1R1 advertised this label asking for the traffic to prefix 33.56.5.5 be labelled with it). So, let's check MPLS forwarding data on SP1R1 for label 101000:

```
RP/0/0/CPU0:SP1R1#show mpls forwarding labels 101000 detail

Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
101000 Exp-Null-v4 33.56.5.5/32       Gi0/0/0/3    33.56.15.5      100233
     Updated: Jan  1 13:41:58.577
     Version: 61, Priority: 3
     Label Stack (Top -> Bottom): { 0 }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 14/18, MTU: 1500
     Packets Switched: 1235
```

We can see that the label will be swapped with label 0 (explicit-null for IPv4) and forwarded to 33.56.5.5 via next hop of 33.56.15.5 over Gi0/0/0/3 interface.



IANA keeps special purpose reserved MPLS label values registry at:
https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xml

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

If you'll need to check remote label mapping for specific prefix (as its often done during troubleshooting) you can check it with the following command – here demonstrated for 12.39.5.5:

```
RP/0/0/CPU0:SP2R2#sh mpls ldp bindings 12.39.5.5/32
12.39.5.5/32, rev 10
        Local binding: label: 123203
        Remote bindings: (3 peers)
            Peer                 Label
            -----------------    ---------
            12.39.1.1:0          123100
            12.39.3.3:0          123300
            12.39.4.4:0          123303
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.2: Explicit NULL label

Some of the features that are in the MPLS network require signalling using the labels carried with packets. With implicit-null configuration, penultimate router will pop all labels before forwarding the packet, which will break those features.

Let's make sure AS3356 uses explicit-null configuration:

```
RP/0/0/CPU0:SP1R1(config)#mpls ldp
RP/0/0/CPU0:SP1R1(config-ldp)#address-family ipv4
RP/0/0/CPU0:SP1R1(config-ldp-af)#label local advertise explicit-null
RP/0/0/CPU0:SP1R1(config-ldp-af-lbl-lcl)#

RP/0/0/CPU0:SP1R2(config)#mpls ldp
RP/0/0/CPU0:SP1R2(config-ldp)#address-family ipv4
RP/0/0/CPU0:SP1R2(config-ldp-af)#label local advertise explicit-null
RP/0/0/CPU0:SP1R2(config-ldp-af)#

RP/0/0/CPU0:SP1R3(config)#mpls ldp
RP/0/0/CPU0:SP1R3(config-ldp)#address-family ipv4
RP/0/0/CPU0:SP1R3(config-ldp-af)#label local advertise explicit-null
RP/0/0/CPU0:SP1R3(config-ldp-af)#

RP/0/0/CPU0:SP1R4(config)#mpls ldp
RP/0/0/CPU0:SP1R4(config-ldp)#address-family ipv4
RP/0/0/CPU0:SP1R4(config-ldp-af)#label local advertise explicit-null
RP/0/0/CPU0:SP1R4(config-ldp-af)#

SP1-R5-RR(config)#mpls ldp explicit-null
```

We can check that we're advertising explicit-null by checking the LDP bindings:

```
RP/0/0/CPU0:SP1R1#sh mpls ldp forwarding

Codes:
  - = GR label recovering, (!) = LFA FRR pure backup path
  {} = Label stack with multi-line output for a routing path
  G = GR, S = Stale, R = Remote LFA FRR backup

Prefix          Label   Label(s)       Outgoing     Next Hop            Flags
                In      Out            Interface                        G S R
--------------- ------- -------------- ------------ ------------------- -----
33.56.2.2/32    101002  ExpNullv4      Gi0/0/0/2    33.56.12.2
33.56.3.3/32    101001  ExpNullv4      Gi0/0/0/5    33.56.13.3
33.56.4.4/32    101003  102003         Gi0/0/0/2    33.56.12.2
                        103003         Gi0/0/0/5    33.56.13.3
33.56.5.5/32    101000  ExpNullv4      Gi0/0/0/3    33.56.15.5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.3: Configuring MPLS TTL propagation

Shut down SP1R5 link to SP1R1:

```
SP1-R5-RR(config)#int gi3
SP1-R5-RR(config-if)#shut
```

Now, with MPLS TTL propagation enabled:

```
CEL7#traceroute 33.56.5.5 source lo0
Type escape sequence to abort.
Tracing the route to 33.56.5.5
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 2 msec 1 msec 1 msec
  2 33.56.23.3 [AS 3356] [MPLS: Label 103000 Exp 0] 6 msec 6 msec 7 msec
  3 33.56.35.5 [AS 3356] 7 msec *  12 msec
```

MPLS TTL propagation is disabled by doing:

```
RP/0/0/CPU0:SP1R1(config)#mpls ip-ttl-propagate disable
RP/0/0/CPU0:SP1R2(config)#mpls ip-ttl-propagate disable
RP/0/0/CPU0:SP1R3(config)#mpls ip-ttl-propagate disable
RP/0/0/CPU0:SP1R4(config)#mpls ip-ttl-propagate disable
SP1-R5-RR(config)#no mpls ip propagate-ttl
```

Now, trace from CEL7 to SP1R5:

```
CEL7#traceroute 33.56.5.5 source lo0
Type escape sequence to abort.
Tracing the route to 33.56.5.5
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 2 msec 1 msec 1 msec
  2 33.56.35.5 [AS 3356] 9 msec *  10 msec
```

Bring the Gi3 link on SP1R5 back again:

```
SP1-R5-RR(config)#int gi3
SP1-R5-RR(config-if)#no shut
```

Enable MPLS IP TTL propagation before continuing to the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.4: Static binding of MPLS labels

Our current routing configuration takes packets to 12.39/16 network using traditional IP forwarding (no MPLS) via SP1R3 from the perspective of AS3356:

```
RP/0/0/CPU0:SP1R1#sh route ipv4 unicast 12.39.0.0
Routing entry for 12.39.0.0/16
  Known via "bgp 3356", distance 200, metric 0
  Tag 1239, type internal
  Installed Jan  1 13:40:38.893 for 05:02:30
  Routing Descriptor Blocks
    33.56.3.3, from 33.56.5.5
      Route metric is 0
  No advertising protos.
```

We can double-check that using traceroute to see the whole path:

```
RP/0/0/CPU0:SP1R1#traceroute 12.39.3.3

Type escape sequence to abort.
Tracing the route to 12.39.3.3

 1  33.56.13.3 0 msec  0 msec  0 msec ! that's the SP1R1<>SP1R3 link
 2  22.22.33.254 0 msec  *  0 msec    ! and that's SP1R3<>SP2R2 link
```

We can also verify in the MPLS forwarding table that no MPLS-based forwarding will take place to this prefix (we assigned labels only to host routes, effectively eliminating this possibility, at least for now):

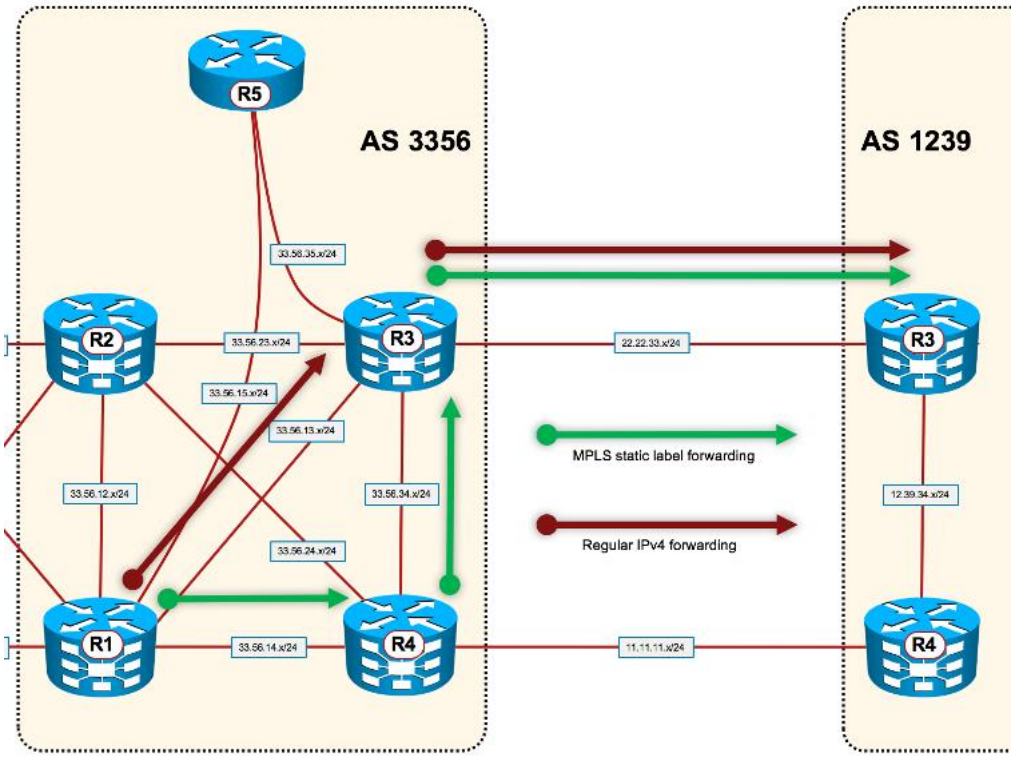
```
RP/0/0/CPU0:SP1R1#sh mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
101000 Exp-Null-v4 33.56.5.5/32       Gi0/0/0/3    33.56.15.5      6628
101001 Exp-Null-v4 33.56.3.3/32       Gi0/0/0/5    33.56.13.3      3702
101002 Exp-Null-v4 33.56.2.2/32       Gi0/0/0/2    33.56.12.2      4667
101003 Exp-Null-v4 33.56.4.4/32       Gi0/0/0/0    33.56.14.4      4731

RP/0/0/CPU0:SP1R1#sh mpls ldp forwarding
Codes:
  - = GR label recovering, (!) = LFA FRR pure backup path
  {} = Label stack with multi-line output for a routing path
  G = GR, S = Stale, R = Remote LFA FRR backup

Prefix           Label   Label(s)       Outgoing     Next Hop            Flags
                 In      Out            Interface                        G S R
---------------- ------- -------------- ------------ ------------------- -----
33.56.2.2/32     101002  ExpNullv4      Gi0/0/0/2    33.56.12.2
33.56.3.3/32     101001  ExpNullv4      Gi0/0/0/5    33.56.13.3
33.56.4.4/32     101003  ExpNullv4      Gi0/0/0/0    33.56.14.4
33.56.5.5/32     101000  ExpNullv4      Gi0/0/0/3    33.56.15.5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

There are multiple ways to distribute labels – using LDP, RSVP, BGP. There's also way to statically define label mappings. We'll make a static LSP that will forward traffic to 12.39.0.0/16 prefix from SP1R1 via SP1R4 and then to SP1R3 for final push (without labels) to SP2R:



On SP1R1 we need to map label to prefix:

```
RP/0/0/CPU0:SP1R1#sh running-config mpls static
mpls static
 address-family ipv4 unicast
  local-label 600 allocate per-prefix 12.39.0.0/16                    ! (1)
   forward
    path 1 nexthop GigabitEthernet0/0/0/0 33.56.14.4 out-label 1432   ! (2)
```

This configuration effectively means for the router:

1. Allocate lobal label of 600 for forwarding to prefix 12.39/16
2. Forward the traffic to this prefix out of GE0/0/0/0 using next hop of 33.56.14.14 (SP1R4 router) with label 1432.

On SP1R2 we already have packet with label 1432, so let's switch it to another interface – towards SP1R3, keeping the same label 1432:

```
RP/0/0/CPU0:SP1R4#sh running-config mpls static
mpls static
 address-family ipv4 unicast
  local-label 1432 allocate
   forward
    path 1 nexthop GigabitEthernet0/0/0/2 33.56.34.3 out-label 1432
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Finally, at the last hop of our AS, we need to pop the label and forward the IP traffic towards AS1239:

```
RP/0/0/CPU0:SP1R3#sh running-config mpls static
mpls static
 address-family ipv4 unicast
  local-label 1432 allocate
   forward
    path 1 nexthop GigabitEthernet0/0/0/1 22.22.33.254 out-label pop
```

To verify, first we'll check SP1R1 MPLS LDP forwarding table versus general MPLS forwarding table – LDP shouldn't have this label allocated, while the general MPLS table should:

```
RP/0/0/CPU0:SP1R1#sh mpls ldp forwarding

Codes:
  - = GR label recovering, (!) = LFA FRR pure backup path
  {} = Label stack with multi-line output for a routing path
  G = GR, S = Stale, R = Remote LFA FRR backup

Prefix          Label   Label(s)       Outgoing     Next Hop            Flags
                In      Out            Interface                        G S R
--------------- ------- -------------- ------------ ------------------- -----
33.56.2.2/32    101002  ExpNullv4      Gi0/0/0/2    33.56.12.2
33.56.3.3/32    101001  ExpNullv4      Gi0/0/0/5    33.56.13.3
33.56.4.4/32    101003  ExpNullv4      Gi0/0/0/0    33.56.14.4
33.56.5.5/32    101000  ExpNullv4      Gi0/0/0/3    33.56.15.5
```

And general MPLS table:

```
RP/0/0/CPU0:SP1R1#sh mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
600    1432        12.39.0.0/16       Gi0/0/0/0    33.56.14.4      0
101000 Exp-Null-v4 33.56.5.5/32       Gi0/0/0/3    33.56.15.5      1089
101001 Exp-Null-v4 33.56.3.3/32       Gi0/0/0/5    33.56.13.3      624
101002 Exp-Null-v4 33.56.2.2/32       Gi0/0/0/2    33.56.12.2      697
101003 Exp-Null-v4 33.56.4.4/32       Gi0/0/0/0    33.56.14.4      835
```

We can also check if the MPLS static label creation was successful:

```
RP/0/0/CPU0:SP1R1#sh mpls static local-label all
Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
600     default         Per-Prefix   12.39.0.0/16     Yes             Created
```



If there's error in creating the label mapping (column **Status** will say '**Error**'), try to check the **show mpls ldp trace** to find a clue)

And finally, we can ping 12.39.3.3 to check if the traffic will go through.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

First, we'll check counters for the prefix on all hops:

```
RP/0/0/CPU0:SP1R1#sh mpls forwarding | utility egrep "Swit|12.39"
Label  Label       or ID              Interface                      Switched
600    1432        12.39.0.0/16       Gi0/0/0/0    33.56.14.4        0
RP/0/0/CPU0:SP1R4#sh mpls forwarding | utility egrep "Swit|1432"
Label  Label       or ID              Interface                      Switched
1432   1432        No ID              Gi0/0/0/2    33.56.34.3        0
RP/0/0/CPU0:SP1R3#sh mpls forwarding | utility egrep "Swit|1432"
Label  Label       or ID              Interface                      Switched
1432   Pop         No ID              Gi0/0/0/1    22.22.33.254      0
```

We'll now traceroute to the 12.39.3.3 (SP2R3 loopback 0 address):

```
RP/0/0/CPU0:SP1R1#traceroute 12.39.3.3 source 33.56.1.1

Type escape sequence to abort.
Tracing the route to 12.39.3.3

 1  33.56.14.4 [MPLS: Label 1432 Exp 0] 19 msec  9 msec  0 msec
 2  33.56.34.3 [MPLS: Label 1432 Exp 0] 9 msec  0 msec  0 msec
 3  22.22.33.254 0 msec  *  0 msec
```

We can also see that the counters reflect traffic movement:

```
RP/0/0/CPU0:SP1R1#sh mpls forwarding | utility egrep "Swit|12.39"
Label  Label       or ID              Interface                      Switched
600    1432        12.39.0.0/16       Gi0/0/0/0    33.56.14.4        288
RP/0/0/CPU0:SP1R4#sh mpls forwarding | utility egrep "Swit|1432"
Label  Label       or ID              Interface                      Switched
1432   1432        No ID              Gi0/0/0/2    33.56.34.3        792
RP/0/0/CPU0:SP1R3#sh mpls forwarding | utility egrep "Swit|1432"
Label  Label       or ID              Interface                      Switched
1432   Pop         No ID              Gi0/0/0/1    22.22.33.254      1134
```

Please remove MPLS static mappings before moving to the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.5: LDP security – authentication

To authenticate LDP you need to specify per-neighbor passwords. Hashes are then used to make sure session is established only with peer that knows the same shared key.

For IOS XR the configuration includes specifying neighbor, label space (for global table it's 0) and password:

```
RP/0/0/CPU0:SP1R1(config)#mpls ldp
RP/0/0/CPU0:SP1R1(config-ldp)#neighbor 33.56.2.2:0 password ?
  clear       Specifies an UNENCRYPTED password will follow
  disable     Disables the global password from this neighbor
  encrypted   Specifies an ENCRYPTED password will follow
RP/0/0/CPU0:SP1R1(config-ldp)#neighbor 33.56.2.2:0 password clear 123
```

And the same for peering router:

```
RP/0/0/CPU0:SP1R2(config)#mpls ldp
RP/0/0/CPU0:SP1R2(config-ldp)#neighbor 33.56.1.1:0 password clear 123
```

Now, the information about used hash (MD5) will be printed next to established neighbor session:

```
RP/0/0/CPU0:SP1R1#sh mpls ldp neighbor

Peer LDP Identifier: 33.56.2.2:0
  TCP connection: 33.56.2.2:40462 - 33.56.1.1:646; MD5 on
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 10/10; Downstream-Unsolicited
  Up time: 00:01:49
  LDP Discovery Sources:
    IPv4: (1)
      GigabitEthernet0/0/0/2
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (5)
      33.56.2.2       33.56.12.2      33.56.23.2      33.56.24.2
      40.60.27.2
    IPv6: (0)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.6: LDP security – label filtering

By default, LDP both assigns labels to all prefixes installed by IGP but also accept label advertisements from other nodes.

To control the accepted bindings, we need to configure an ACL with acceptable prefixes and specify neighbor by its LDP ID or IP address.

Let's add to SP1R5 two additional interfaces and include them in OSPF instance to be picked up by LDP to advertise:

```
SP1-R5-RR(config)#int lo10
SP1-R5-RR(config-if)#ip address 33.56.10.10 255.255.255.255
SP1-R5-RR(config-if)#ip ospf 100 area 0
SP1-R5-RR(config-if)#int lo20
SP1-R5-RR(config-if)#ip address 33.56.20.20 255.255.255.255
SP1-R5-RR(config-if)#ip ospf 100 area 0
SP1-R5-RR(config-if)#^Z
SP1-R5-RR#
```

Let's check current accepted advertisements on SP1R2:

```
RP/0/0/CPU0:SP1R2#sh mpls ldp forwarding

Codes:
  - = GR label recovering, (!) = LFA FRR pure backup path
  {} = Label stack with multi-line output for a routing path
  G = GR, S = Stale, R = Remote LFA FRR backup

Prefix          Label   Label(s)       Outgoing     Next Hop            Flags
                In      Out            Interface                        G S R
--------------- ------- -------------- ------------ ------------------- -----
33.56.1.1/32    102001  ExpNullv4      Gi0/0/0/2    33.56.12.1
33.56.3.3/32    102000  ExpNullv4      Gi0/0/0/0    33.56.23.3
33.56.4.4/32    102003  ExpNullv4      Gi0/0/0/4    33.56.24.4
33.56.5.5/32    102002  103000         Gi0/0/0/0    33.56.23.3
                        101000         Gi0/0/0/2    33.56.12.1
33.56.10.10/32  102009  103004         Gi0/0/0/0    33.56.23.3
                        101008         Gi0/0/0/2    33.56.12.1
33.56.20.20/32  102010  103005         Gi0/0/0/0    33.56.23.3
                        101009         Gi0/0/0/2    33.56.12.1
```

We can see that apart from our normal loopbacks, two new IPs appeared in the routing table and LDP assigned labels to them, as they're both host routes.

Now, let's build a filter and make sure from 33.56.3.3 we'll accept only specific mapping – that of 33.56.5.5/32:

```
RP/0/0/CPU0:SP1R2(config)#ipv4 access-list R5-LOOP
RP/0/0/CPU0:SP1R2(config-ipv4-acl)#permit ipv4 host 33.56.5.5 any
RP/0/0/CPU0:SP1R2(config-ipv4-acl)#commit
RP/0/0/CPU0:SP1R2(config)#mpls ldp
RP/0/0/CPU0:SP1R2(config-ldp)#address-family ipv4
RP/0/0/CPU0:SP1R2(config-ldp-af)#label remote accept from 33.56.3.3:0 for R5-LOOP
RP/0/0/CPU0:SP1R2(config-ldp-af)#exit
RP/0/0/CPU0:SP1R2(config-ldp)#exit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

After configuration we can see that the label mappings for 33.56.10.10/32 and 33.56.20.20/32 were removed as advertised by SP1R3, despite the fact, that SP1R3 can still advertise us label mapping for 33.56.5.5:

```
RP/0/0/CPU0:SP1R2#sh mpls ldp forwarding
[…]
Prefix           Label   Label(s)       Outgoing     Next Hop            Flags
                 In      Out            Interface                        G S R
---------------- ------- -------------- ------------ ------------------- -----
33.56.5.5/32     102002  103000         Gi0/0/0/0    33.56.23.3
                         101000         Gi0/0/0/2    33.56.12.1
33.56.10.10/32   102009  Unlabelled     Gi0/0/0/0    33.56.23.3
                         101008         Gi0/0/0/2    33.56.12.1
33.56.20.20/32   102010  Unlabelled     Gi0/0/0/0    33.56.23.3
                         101009         Gi0/0/0/2    33.56.12.1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.7: Turning on MPLS in AS1239 and AS200

To finish the exercise, turn on MPLS forwarding using LDP protocol in AS1239 and AS200 with following assumptions:

- enable explicit-null labels
- leave the MPLS TTL propagation on
- enable MPLS LDP sync
- label ranges:
  - SP2R1: 123100 – 123199
  - SP2R2: 123200 – 123299
  - SP2R3: 123300 – 123399
  - SP2R4: 123400 – 123499
  - SP2R5: 123500 – 123599

MPLS LDP sync in IS-IS on IOS XR is configured under AF config of specific interface, like that (for example):

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#router isis AS1239
RP/0/0/CPU0:SP2R2(config-isis)#interface GigabitEthernet0/0/0/0
RP/0/0/CPU0:SP2R2(config-isis-if)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R2(config-isis-if-af)#mpls ldp sync
RP/0/0/CPU0:SP2R2(config-isis-if-af)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3: Basic MPLS OAM

Task 3.1: Turning on MPLS OAM for IOS XR

Configuration is pretty straightforward – after enabling MPLS on a router, enable mpls oam in global configuration mode:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#mpls oam
RP/0/0/CPU0:SP1R2(config-oam)#commit
```

Please repeat this for all IOS XR routers in AS 3356.

Task 3.2: Turning on MPLS OAM for IOS XE

Configuration is pretty straightforward as well, and executed in the same way – just activate mpls oam in global configuration mode:

```
SP1R5-RR#conf t
SP1R5-RR(config)#mpls oam
SP1R5-RR(config-mpls)#^Z
SP1R5-RR#
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.3: Using MPLS ping and traceroute on IOS XE

Once MPLS OAM is activated you can ping and trace using LSP with addition of ‘mpls’ into ping and traceroute.

```
SP1R5-RR#traceroute mpls ipv4 33.56.2.2/32
Tracing MPLS Label Switched Path to 33.56.2.2/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'l' - Label switched with FEC change, 'd' - see DDMAP for return code,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
  0 33.56.15.5 MRU 1500 [Labels: 24003 Exp: 0]
L 1 33.56.15.1 MRU 1500 [Labels: implicit-null Exp: 0] 4 ms
! 2 33.56.12.2 8 ms

SP1R5-RR#ping mpls ipv4 33.56.2.2/32
Sending 5, 72-byte MPLS Echos to 33.56.2.2/32,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'l' - Label switched with FEC change, 'd' - see DDMAP for return code,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/5/6 ms
 Total Time Elapsed 39 ms
```

Note the extended codes returned by those commands.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.4. Using MPLS ping and traceroute on IOS XR

Once MPLS OAM is activated you can ping and trace using LSP with addition of ‘mpls’ into ping and traceroute.

```
RP/0/0/CPU0:SP1R2#traceroute mpls ipv4 33.56.5.5/32
Tracing MPLS Label Switched Path to 33.56.5.5/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

  0 33.56.12.2 MRU 1500 [Labels: 24001 Exp: 0]
L 1 33.56.12.1 MRU 1500 [Labels: implicit-null Exp: 0] 0 ms
! 2 33.56.15.5 10 ms
```

And MPLS ping:

```
RP/0/0/CPU0:SP1R2#ping mpls ipv4 33.56.5.5/32
Sending 5, 100-byte MPLS Echos to 33.56.5.5/32,
      timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/10 ms
```

For IPv6 FECs the operation is similar:

```
RP/0/0/CPU0:SP1R2#ping mpls ipv6 2001:33:56:aa::5/128

Sending 5, 100-byte MPLS Echos to 2001:33:56:aa::5/128,
      timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

BBBBB
Success rate is 0 percent (0/5)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Note the output – SP1R5 is not IPv6 capable LDP peer, so pinging it is not possible. With traceroute however:

```
RP/0/0/CPU0:SP1R2#traceroute mpls ipv6 2001:33:56:aa::5/128
Sun Dec  2 14:15:01.112 UTC

Tracing MPLS Label Switched Path to 2001:33:56:aa::5/128, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

  0 2001:33:56:23::2 MRU 1500 [Labels: 24004 Exp: 0]
B 1 2001:33:56:23::3 MRU 1500 [No Label] 10 ms
```

…we at least reach the penultimate hop and get the information there's no label assigned by R5 to R3 for forwarding information using IPv6.

Tracing route to IPv6 enabled LDP peers works correctly:

```
RP/0/0/CPU0:SP1R2#traceroute mpls ipv6 2001:33:56:aa::4/128

Tracing MPLS Label Switched Path to 2001:33:56:aa::4/128, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

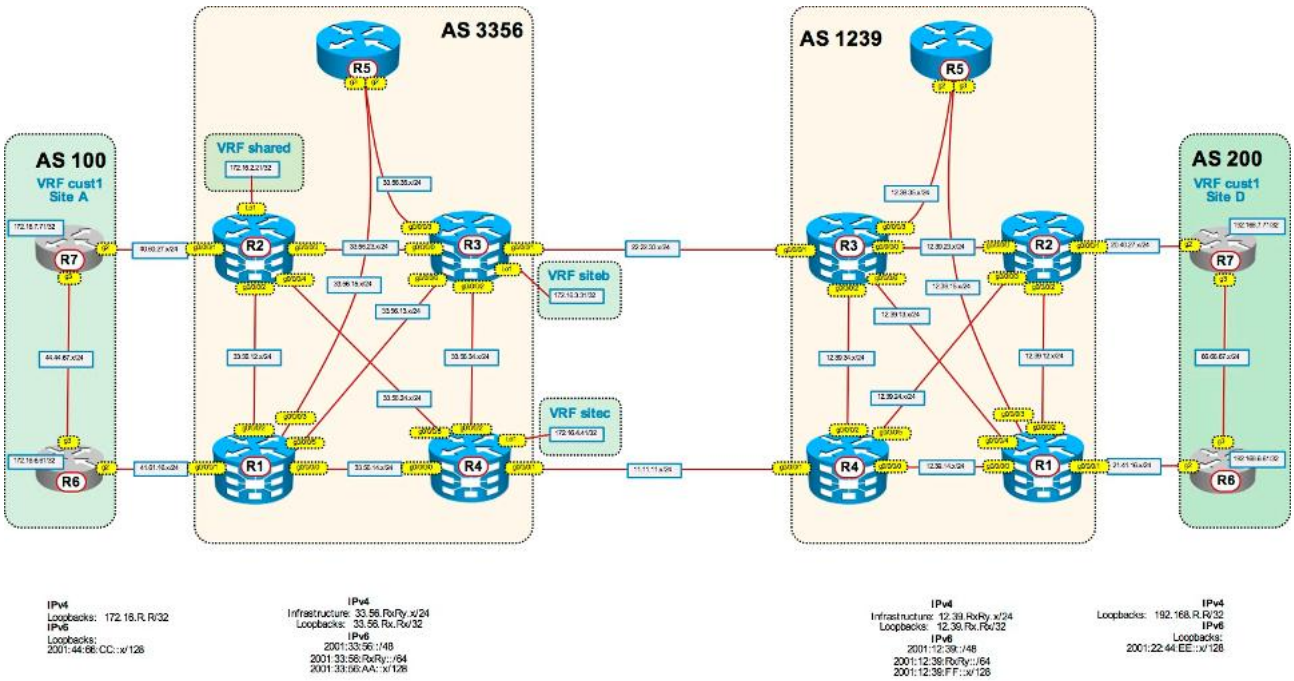
Type escape sequence to abort.

  0 2001:33:56:24::2 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 2001:33:56:24::4 10 ms
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 8:
MPLS Intra-AS L3VPN

In this lab you will build the L3VPN services in AS 3356 and AS 1239.



This chapter uses following init files:

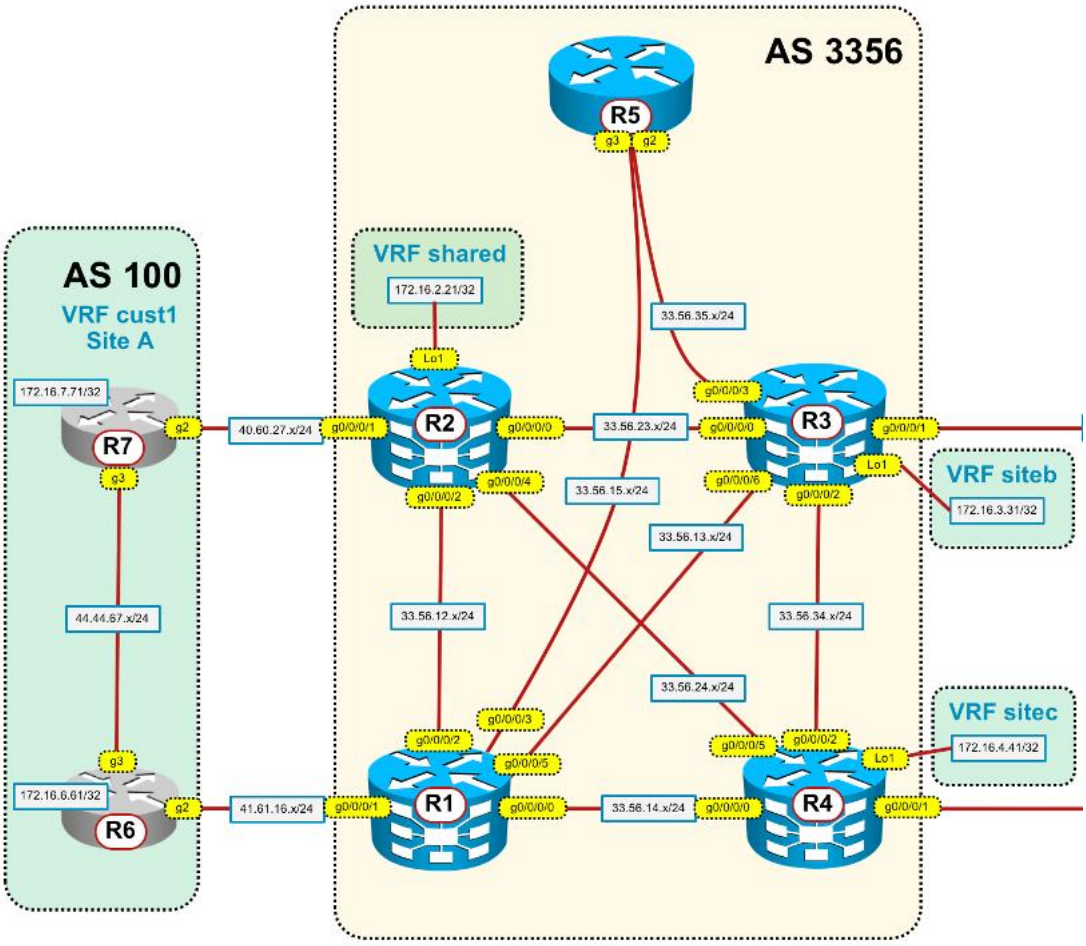
- **For VIRL/CML 1.x:** `sp.bootcamp.ch08.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch08.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch08.unl`

This section is required for the next step which is Inter-AS L3VPN.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: L3VPN in AS 3356

A customer named cust1 is located in AS 100 Site A. Connect this site to the L3VPN service offered by the Service Provider AS 3356. Then add sites B, C and shared services and allow communication between sites.



Task 1.1: Enable VPNv4 and connect Customer1 in Site A

Configure VRF cust1 on R1 and R2 in AS3356.

```
RP/0/0/CPU0:SP1R2(config)#vrf cust1
RP/0/0/CPU0:SP1R2(config-vrf)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-vrf-af)#import route-target
RP/0/0/CPU0:SP1R2(config-vrf-import-rt)#1:100
RP/0/0/CPU0:SP1R2(config-vrf-import-rt)#export route-target
RP/0/0/CPU0:SP1R2(config-vrf-export-rt)#1:100
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Assign the VRF cust1 to the interface.

```
RP/0/0/CPU0:SP1R2(config-vrf-export-rt)#root
RP/0/0/CPU0:SP1R2(config)#int g0/0/0/1
RP/0/0/CPU0:SP1R2(config-if)#vrf cust1
```

Don't commit this configuration yet.

Configure VRF and RD in the BGP process.

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#vrf cust1

RP/0/0/CPU0:SP1R2(config-bgp-vrf)#rd ?
  <1-65535>.<0-65535>:<0-65535>  4-byte AS number in asdot (X.Y) format
  <1-65535>:                     2-byte AS number
  <65536-4294967295>:            4-byte AS number in asplain format
  A.B.C.D:                       IPv4 address
  auto                           Automatic route distinguisher

RP/0/0/CPU0:SP1R2(config-bgp-vrf)#rd 111:100
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#add ipv4 uni
```

Move the eBGP neighbor from the global to VRF instance. Find the neighbor's IP with the '**do sh bgp summary**' and then:

```
RP/0/0/CPU0:SP1R2(config-bgp-vrf-af)#exit
RP/0/0/CPU0:SP1R2(config-bgp-vrf-af)#do sh run router bgp 3356 nei 40.60.27.7

router bgp 3356
 neighbor 40.60.27.7
  remote-as 100
  password encrypted 0028362030
  address-family ipv4 unicast
   route-policy PASS in
   route-policy PASS out
  !
 !
```

Copy and paste the neighbor configuration to the VRF section in BGP.

```
RP/0/0/CPU0:SP1R2(config-bgp-vrf-af)#neighbor 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr)#  remote-as 100
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr)#  password encrypted 0028362030
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr)#  address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr-af)#   route-policy PASS in
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr-af)#   route-policy PASS out
```

Enable the VPNv4 address family to the route-reflector.

```
RP/0/0/CPU0:SP1R2(config-bgp)#nei 33.56.5.5 add vpnv4 uni
```

You can verify a configuration before the commit.

```
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#show config

Building configuration...
!! IOS XR Configuration 6.1.2
vrf cust1
 address-family ipv4 unicast
  import route-target
   1:100
  !
  export route-target
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
   1:100
  !
 !
!
interface GigabitEthernet0/0/0/1
 vrf cust1
!
router bgp 3356
 neighbor 33.56.5.5
  address-family vpnv4 unicast
  !
 !
 vrf cust1
  rd 111:100
  address-family ipv4 unicast
  !
  neighbor 40.60.27.7
   remote-as 100
   password encrypted 0028362030
   address-family ipv4 unicast
    route-policy PASS in
    route-policy PASS out
   !
  !
 !
!

RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#commit

% Failed to commit one or more configuration items during a pseudo-atomic operation. All
changes made have been reverted. Please issue 'show configuration failed [inheritance]'
from this session to view the errors
```

Let's see what's happened.

```
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#show config fail

!! SEMANTIC ERRORS: This configuration was rejected by
!! the system due to semantic errors. The individual
!! errors with each failed configuration command can be
!! found below.


router bgp 3356
 neighbor 33.56.5.5
  address-family vpnv4 unicast
!!% 'BGP' detected the 'warning' condition 'The address family has not been initialized'
  !
 !
!
end
```



One of the typical mistakes is trying to run L3VPN without initializing the address-family VPNv4. It is not enough to enable VPNv4 under the neighbor statement. VPNv4 must be configured also under the global BGP process.

```
RP/0/0/CPU0:SP1R2(config-bgp)# address-family vpnv4 unicast
```

The VPNv4 AFI is required whenever there is the IPv4 AFI enabled in VRF. The same rule applies for VPNv6 and IPv6.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#exit
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#exit
RP/0/0/CPU0:SP1R2(config-bgp)#address-family vpnv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-af)#commit

% Failed to commit one or more configuration items during a pseudo-atomic operation. All
changes made have been reverted. Please issue 'show configuration failed [inheritance]'
from this session to view the errors

bgp[1052]: %ROUTING-BGP-3-RIB_FAIL : [22] : Failed to flush nexthop registrations for
while calling the RIB instance 0: 'Subsystem(1445)' detected the 'warning' condition
'Code(12)'
bgp[1052]: %ROUTING-BGP-3-RIB_FAIL : [22] : Failed to flush nexthop registrations for
while calling the RIB instance 0: 'Subsystem(1445)' detected the 'warning' condition
'Code(12)'

RP/0/0/CPU0:SP1R2(config-bgp-af)#show config fail

!! SEMANTIC ERRORS: This configuration was rejected by
!! the system due to semantic errors. The individual
!! errors with each failed configuration command can be
!! found below.

interface GigabitEthernet0/0/0/1
 vrf cust1
!!% 'RSI' detected the 'fatal' condition 'The interface's numbered and unnumbered
IPv4/IPv6 addresses must be removed prior to changing or deleting the VRF'
!
end
```



Another mistake is to try applying a VRF to the interface without the IP addresses removal. Please remember to add those IP address back again. In IOS XE the IP addresses are removed automatically. In IOS XR you need to remove them manually. Always check the interface configuration beforehand.

Check the interface configuration. Maybe there is an another VRF or there is an indication that this interface should not be within VRF?

```
RP/0/0/CPU0:SP1R2(config-bgp-af)#root
RP/0/0/CPU0:SP1R2(config)#do sh run int g0/0/0/1
interface GigabitEthernet0/0/0/1
 description to CEL7
 ipv4 address 40.60.27.2 255.255.255.0
 ipv6 address fe80::2 link-local
 ipv6 address 2001:40:60:27::2/64
!
```

You can remove IP address, add a VRF and re-address the interface in one config cycle.

```
RP/0/0/CPU0:SP1R2(config)#int GigabitEthernet0/0/0/1
RP/0/0/CPU0:SP1R2(config-if)#no ipv4 add
RP/0/0/CPU0:SP1R2(config-if)#no ipv6 add
RP/0/0/CPU0:SP1R2(config-if)#vrf cust1
RP/0/0/CPU0:SP1R2(config-if)#ipv4 address 40.60.27.2 255.255.255.0
RP/0/0/CPU0:SP1R2(config-if)#ipv6 address fe80::2 link-local
RP/0/0/CPU0:SP1R2(config-if)#ipv6 address 2001:40:60:27::2/64
RP/0/0/CPU0:SP1R2(config-if)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

One you move the interface to the VRF on R1/R2 the following message may appear.

```
%TCP-6-BADAUTH: No MD5 digest
```

This is because R6 and R7 are trying to establish an IPv6 eBGP session. You can disable it by removing the 'activate' line from the 'address-family ipv6 unicast' on R6 and R7. Bring this configuration back in the Task 2 where you enable L3VPN services for IPv6.

Enable VPNv4 on RR. Configure the route-reflector for all routers in the AS3356.

```
SP1-R5-RR(config)#router bgp 3356
SP1-R5-RR(config-router)#add vpnv4 uni
SP1-R5-RR(config-router-af)#neighbor 33.56.1.1 activate
SP1-R5-RR(config-router-af)#neighbor 33.56.1.1 route-reflector-client
SP1-R5-RR(config-router-af)#neighbor 33.56.2.2 activate
SP1-R5-RR(config-router-af)#neighbor 33.56.2.2 route-reflector-client
SP1-R5-RR(config-router-af)#neighbor 33.56.3.3 activate
SP1-R5-RR(config-router-af)#neighbor 33.56.3.3 route-reflector-client
SP1-R5-RR(config-router-af)#neighbor 33.56.4.4 activate
SP1-R5-RR(config-router-af)#neighbor 33.56.4.4 route-reflector-client
```

Verification of VPNv4 R2-RR after configuring of R2 and R5 (RR):

```
RP/0/0/CPU0:SP1R2#sh ip vrf all int brief

Interface                      IP-Address      Status          Protocol Vrf-Name
Loopback0                      33.56.2.2       Up              Up       default
MgmtEth0/0/CPU0/0              10.255.0.201    Up              Up       Mgmt-intf
GigabitEthernet0/0/0/0         33.56.23.2      Up              Up       default
GigabitEthernet0/0/0/1         40.60.27.2      Up              Up       cust1
GigabitEthernet0/0/0/2         33.56.12.2      Up              Up       default
GigabitEthernet0/0/0/3         41.61.26.2      Shutdown        Down     default
GigabitEthernet0/0/0/4         33.56.24.2      Up              Up       default

RP/0/0/CPU0:SP1R2#sh bgp vrf cust1 summ
BGP VRF cust1, state: Active
BGP Route Distinguisher: 111:100
VRF ID: 0x60000005
BGP router identifier 33.56.2.2, local AS number 3356
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000014   RD version: 7
BGP main routing table version 7
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0

BGP is operating in STANDALONE mode.


Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker               7          7          7          7           7           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
40.60.27.7        0   100    2466    1316        7    0    0 21:54:23          3
```

There are 3 prefixes received from the neighbor.

```
RP/0/0/CPU0:SP1R2#sh route vrf cust1
[..]

Gateway of last resort is not set
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
C    40.60.27.0/24 is directly connected, 21:54:39, GigabitEthernet0/0/0/1
L    40.60.27.2/32 is directly connected, 21:54:39, GigabitEthernet0/0/0/1
B    44.44.67.0/24 [20/0] via 40.60.27.7, 21:54:08
B    172.16.6.6/32 [20/0] via 40.60.27.7, 21:54:08
B    172.16.7.7/32 [20/0] via 40.60.27.7, 21:54:08


RP/0/0/CPU0:SP1R2#sh bgp vpnv4 unicast summ
[..]
Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
33.56.5.5         0  3356    2849    1649        7    0    0 21:57:48          0
```

Verify advertised routes to RR.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 unicast nei 33.56.5.5 advertised-routes
Network            Next Hop        From            AS Path
Route Distinguisher: 111:100
44.44.67.0/24      33.56.2.2       40.60.27.7      100i
172.16.6.6/32      33.56.2.2       40.60.27.7      100i
172.16.7.7/32      33.56.2.2       40.60.27.7      100i

Processed 3 prefixes, 3 paths
```

There are default BGP attributes set with advertised prefixes.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 unicast
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100 (default for vrf cust1)
*> 44.44.67.0/24      40.60.27.7               0             0 100 i
*> 172.16.6.6/32      40.60.27.7                             0 100 i
*> 172.16.7.7/32      40.60.27.7               0             0 100 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: Add R1, R3 and R4 to L3VPN

Configure VRF cust1 on R1 respectively to R2's settings in the Task 1. Enable the VPNv4 AFI on R3 and R4 (no VRFs there). Verification after the configuration is done.

```
SP1-R5-RR#sh ip bgp vpnv4 all summ
BGP router identifier 33.56.5.5, local AS number 3356
BGP table version is 13, main routing table version 13
3 network entries using 768 bytes of memory
6 path entries using 768 bytes of memory
2/2 BGP path/bestpath attribute entries using 576 bytes of memory
3 BGP AS-PATH entries using 88 bytes of memory
1 BGP extended community entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2224 total bytes of memory
BGP activity 1519/1506 prefixes, 1574/1544 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
33.56.1.1       4         3356      17      26       13    0    0 00:07:35        3
33.56.2.2       4         3356    1341    2499       13    0    0 22:16:46        3
33.56.3.3       4         3356       7      12       13    0    0 00:00:18        0
33.56.4.4       4         3356       7       9       13    0    0 00:00:05        0


SP1-R5-RR#sh ip bgp vpnv4 all
BGP table version is 13, local router ID is 33.56.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100
 *>i  44.44.67.0/24    33.56.1.1                0    100      0 100 i
 * i                   33.56.2.2                0    100      0 100 i
 *>i  172.16.6.6/32    33.56.1.1                0    100      0 100 i
 * i                   33.56.2.2                     100      0 100 i
 *>i  172.16.7.7/32    33.56.1.1                     100      0 100 i
 * i                   33.56.2.2                0    100      0 100 i
```

RR has two paths for the same IPv4 route.

```
SP1-R5-RR#sh ip bgp vpnv4 all 172.16.6.6
BGP routing table entry for 111:100:172.16.6.6/32, version 12
Paths: (2 available, best #1, no table)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  100, (Received from a RR-client)
    33.56.1.1 (metric 2) (via default) from 33.56.1.1 (33.56.1.1)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:100
      mpls labels in/out nolabel/101005
      rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 1
  100, (Received from a RR-client)
    33.56.2.2 (metric 3) (via default) from 33.56.2.2 (33.56.2.2)
      Origin IGP, localpref 100, valid, internal
      Extended Community: RT:1:100
      mpls labels in/out nolabel/102005
      rx pathid: 0, tx pathid: 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3: Additional paths with Unique RD

Why R2 has 6 paths and R1 just 3 paths?

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 unicast
[...]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100 (default for vrf cust1)
* i44.44.67.0/24      33.56.1.1                0    100      0 100 i
*>                    40.60.27.7               0             0 100 i
* i172.16.6.6/32      33.56.1.1                0    100      0 100 i
*>                    40.60.27.7                             0 100 i
* i172.16.7.7/32      33.56.1.1                     100      0 100 i
*>                    40.60.27.7               0             0 100 i

Processed 3 prefixes, 6 paths


RP/0/0/CPU0:SP1R1#sh bgp vpnv4 unicast
[...]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100 (default for vrf cust1)
*> 44.44.67.0/24      41.61.16.6               0             0 100 i
*> 172.16.6.6/32      41.61.16.6               0             0 100 i
*> 172.16.7.7/32      41.61.16.6                             0 100 i

Processed 3 prefixes, 3 paths
```

This is because RR sends only the best path by default to its clients.

Configure routers to advertise additional 3 paths also to R1 without using Add-Path. To make this happen you can use the Unique-RD technique. Advertise the same prefixes with two different RDs. R1 is sending prefixes with RD 111:100. R2 will have RD set to 112:100.

The current settings:

```
router bgp 3356
 vrf cust1
  rd 111:100
  address-family ipv4 unicast
  !
  neighbor 40.60.27.7
   remote-as 100
   password encrypted 0028362030
   address-family ipv4 unicast
    route-policy PASS in
    route-policy PASS out
   !
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Change RD on R2 from 111:100 to 112:100.

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#vrf cust1
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#rd 112:100

% Failed to commit one or more configuration items during a pseudo-atomic operation. All
changes made have been reverted. Please issue 'show configuration failed [inheritance]'
from this session to view the errors
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#sh config failed
!! SEMANTIC ERRORS: This configuration was rejected by
!! the system due to semantic errors. The individual
!! errors with each failed configuration command can be
!! found below.

router bgp 3356
 vrf cust1
  rd 112:100
!!% 'BGP' detected the 'warning' condition 'The RD cannot be changed or removed while a
VRF address family is active'
 !
!
```

You need to remove a neighbor which is using this RD.

```
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#no add ipv4 uni
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#no nei 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#rd 112:100
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#commit
```

Paste the config of a BGP neighbor to configure the neighbor back again.

```
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#add ipv4 uni
RP/0/0/CPU0:SP1R2(config-bgp-vrf-af)#  neighbor 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr)#   remote-as 100
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr)#   password encrypted 0028362030
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr)#   address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr-af)#    route-policy PASS in
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr-af)#    route-policy PASS out
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr-af)#   !
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr-af)#commit
```

Verification. Wait a while. R1 and R2 should have 6 paths in VRF with their native RD as they are imported from a different VPN (RD). If you don't have the same output, you can clear bgp sessions by '**clear bgp vpnv4 unicast * soft in|out**'.

```
SP1-R5-RR#sh ip bgp vpnv4 all
BGP table version is 16, local router ID is 33.56.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100
 *>i  44.44.67.0/24    33.56.1.1                0    100      0 100 i
 *>i  172.16.6.6/32    33.56.1.1                0    100      0 100 i
 *>i  172.16.7.7/32    33.56.1.1                     100      0 100 i
Route Distinguisher: 112:100
 *>i  44.44.67.0/24    33.56.2.2                0    100      0 100 i
 *>i  172.16.6.6/32    33.56.2.2                     100      0 100 i
 *>i  172.16.7.7/32    33.56.2.2                0    100      0 100 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

On RR there are 6 different paths for 3 IPv4 prefixes.

Verify the BGP table on R2 and R1.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 unicast
[..]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100
*>i44.44.67.0/24      33.56.1.1                0    100      0 100 i
*>i172.16.6.6/32      33.56.1.1                0    100      0 100 i
*>i172.16.7.7/32      33.56.1.1                     100      0 100 i
Route Distinguisher: 112:100 (default for vrf cust1)
* i44.44.67.0/24      33.56.1.1                0    100      0 100 i
*>                    40.60.27.7               0             0 100 i
* i172.16.6.6/32      33.56.1.1                0    100      0 100 i
*>                    40.60.27.7                             0 100 i
* i172.16.7.7/32      33.56.1.1                     100      0 100 i
*>                    40.60.27.7               0             0 100 i

Processed 6 prefixes, 9 paths


RP/0/0/CPU0:SP1R1#sh bgp vpnv4 unicast
[..]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100 (default for vrf cust1)
* i44.44.67.0/24      33.56.2.2                0    100      0 100 i
*>                    41.61.16.6               0             0 100 i
* i172.16.6.6/32      33.56.2.2                     100      0 100 i
*>                    41.61.16.6               0             0 100 i
* i172.16.7.7/32      33.56.2.2                0    100      0 100 i
*>                    41.61.16.6                             0 100 i
Route Distinguisher: 112:100
*>i44.44.67.0/24      33.56.2.2                0    100      0 100 i
*>i172.16.6.6/32      33.56.2.2                     100      0 100 i
*>i172.16.7.7/32      33.56.2.2                0    100      0 100 i

Processed 6 prefixes, 9 paths
```

From the L3VPN's network perspective there are two VPNs. One with RD 111:100 and one with RD 112:100. The VRF name is locally significant only.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4: Provide redundancy to site A

When the link CEL7-CEL6 fails the prefixes from cust1 should be still reachable. To allow prefixes to be re-advertised back to AS 100 the as-override or allowas-in features must be enabled. In this case there is as-override.

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#vrf cust1
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#nei 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr)#add ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr-af)#as-override
```

Do the same on R1.

Verification before link CEL7-CEL6 is shut down.

```
CEL7#sh ip route 172.16.6.6
Routing entry for 172.16.6.6/32
  Known via "ospf 100", distance 110, metric 301, type intra area
  Last update from 44.44.67.6 on GigabitEthernet3, 6d09h ago
  Routing Descriptor Blocks:
  * 44.44.67.6, from 172.16.6.6, 6d09h ago, via GigabitEthernet3
      Route metric is 301, traffic share count is 1
```

The interface Gi3 is up and running. This is the outgoing interface.

```
CEL7#sh ip int brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       10.255.0.197    YES TFTP   up                    up
GigabitEthernet2       40.60.27.7      YES TFTP   up                    up
GigabitEthernet3       44.44.67.7      YES TFTP   up                    up
GigabitEthernet4       22.42.37.7      YES TFTP   up                    up
GigabitEthernet5       40.60.17.7      YES TFTP   administratively down down
Loopback0              172.16.7.7      YES TFTP   up                    up
```

Disable Gi3.

```
CEL7(config)#int g3
CEL7(config-if)#shut

CEL7(config-if)#do sh ip route 172.16.6.6
Routing entry for 172.16.6.6/32
  Known via "bgp 100", distance 200, metric 0, type internal
  Last update from 172.16.6.6 00:00:05 ago
  Routing Descriptor Blocks:
  * 172.16.6.6, from 172.16.6.6, 00:00:05 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
      MPLS label: none
```

It seems fine but there is a recursive lookup loop. The BGP session CEL7-CEL6 is flapping.

```
* 10:17:09.500: %BGP-3-NOTIFICATION: sent to neighbor 172.16.6.6 4/0 (hold time expired)
0 bytes
* 10:17:09.501: %BGP-5-NBR_RESET: Neighbor 172.16.6.6 reset (BGP Notification sent)
* 10:17:09.501: %BGP-5-ADJCHANGE: neighbor 172.16.6.6 Down BGP Notification sent
* 10:17:09.502: %BGP_SESSION-5-ADJCHANGE: neighbor 172.16.6.6 IPv4 Unicast topology base
removed from session  BGP Notification sent
CEL7#
* 10:17:19.372: %BGP-5-ADJCHANGE: neighbor 172.16.6.6 Up
```

When the direct iBGP session is down then the prefix is learnt correctly.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
* 10:20:29.234: %BGP-3-NOTIFICATION: sent to neighbor 172.16.6.6 4/0 (hold time expired)
0 bytes
%BGP-5-NBR_RESET: Neighbor 172.16.6.6 reset (BGP Notification sent)
%BGP-5-ADJCHANGE: neighbor 172.16.6.6 Down BGP Notification sent
%BGP_SESSION-5-ADJCHANGE: neighbor 172.16.6.6 IPv4 Unicast topology base removed from
session  BGP Notification sent
```

Periodically the communication is re-established but just until the next flapping.

```
CEL7#sh ip route
[..]
B        44.44.67.0 [20/0] via 40.60.27.2, 00:00:03
      172.16.0.0/32 is subnetted, 2 subnets
B        172.16.6.6 [20/0] via 40.60.27.2, 00:00:03
C        172.16.7.7 is directly connected, Loopback0

CEL7#ping 172.16.6.6 sou lo0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.6.6, timeout is 2 seconds:
Packet sent with a source address of 172.16.7.7
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 11/11/12 ms
CEL7#
```

Add an additional Loopback1 and advertise it instead of Loopback0 to avoid a recursive loop.

```
CEL7(config)#int lo1
CEL7(config-if)#ip add 172.16.7.71 255.255.255.255
CEL7(config-if)#no ip prefix-list PFX-RED-CONN permit 172.16.7.7/32
CEL7(config-if)#ip prefix-list PFX-RED-CONN permit 172.16.7.71/32

CEL6(config)#int lo1
CEL6(config-if)#ip add 172.16.6.61 255.255.255.255
CEL6(config-if)#no ip prefix-list PFX-RED-CONN permit 172.16.6.6/32
CEL6(config-if)#ip prefix-list PFX-RED-CONN permit 172.16.6.61/32
```

Now it works fine.

```
CEL7#sh ip route 172.16.6.61
Routing entry for 172.16.6.61/32
  Known via "bgp 100", distance 20, metric 0
  Tag 3356, type external
  Last update from 40.60.27.2 00:00:47 ago
  Routing Descriptor Blocks:
  * 40.60.27.2, from 40.60.27.2, 00:00:47 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2
      Route tag 3356
      MPLS label: none

CEL7#ping 172.16.6.61 sou lo1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.6.61, timeout is 2 seconds:
Packet sent with a source address of 172.16.7.71
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 11/12/13 ms

CEL7#trace 172.16.6.61 sou lo1
Type escape sequence to abort.
Tracing the route to 172.16.6.61
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 2 msec 1 msec 2 msec
  2 33.56.12.1 [MPLS: Labels 0/101012 Exp 0] 46 msec 13 msec 11 msec
  3 41.61.16.6 11 msec *  14 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Routes are learnt.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 uni
[..]

   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100
*>i44.44.67.0/24      33.56.1.1                0    100      0 100 i
*>i172.16.6.61/32     33.56.1.1                0    100      0 100 i
Route Distinguisher: 112:100 (default for vrf cust1)
*>i44.44.67.0/24      33.56.1.1                0    100      0 100 i
*>i172.16.6.61/32     33.56.1.1                0    100      0 100 i
*> 172.16.7.71/32     40.60.27.7               0             0 100 i

Processed 5 prefixes, 5 paths
```

Bring the Gi3 interface back and verify the BGP tables.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 uni
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100
*>i44.44.67.0/24      33.56.1.1                0    100      0 100 i
*>i172.16.6.61/32     33.56.1.1                0    100      0 100 i
*>i172.16.7.71/32     33.56.1.1                     100      0 100 i
Route Distinguisher: 112:100 (default for vrf cust1)
* i44.44.67.0/24      33.56.1.1                0    100      0 100 i
*>                    40.60.27.7               0             0 100 i
* i172.16.6.61/32     33.56.1.1                0    100      0 100 i
*>                    40.60.27.7                             0 100 i
* i172.16.7.71/32     33.56.1.1                     100      0 100 i
*>                    40.60.27.7               0             0 100 i

Processed 6 prefixes, 9 paths
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.5: Add two additional VRF sites.

Add two additional VRFs siteb and sitec to emulate two locations. Configure Loopback1 on R3 and R4 with the following settings.

R3:

IP address of Loopback1: 172.16.3.31/32

VRF name: siteb

RD: 3:3

RT: 3:3

R4:

IP address of Loopback1: 172.16.4.41/32

VRF name: sitec

RD: 4:4

RT: 4:4

As the result R6 and R7 should be able to communicate with R3's Lo1 and R4's Lo1. But VRFs siteb and sitec should not be aware of each other prefixes.

```
RP/0/0/CPU0:SP1R3(config)#vrf siteb
RP/0/0/CPU0:SP1R3(config-vrf)# address-family ipv4 unicast
RP/0/0/CPU0:SP1R3(config-vrf-af)#  import route-target
RP/0/0/CPU0:SP1R3(config-vrf-import-rt)#   3:3
RP/0/0/CPU0:SP1R3(config-vrf-import-rt)#   1:100
RP/0/0/CPU0:SP1R3(config-vrf-import-rt)#  export route-target
RP/0/0/CPU0:SP1R3(config-vrf-export-rt)#   3:3
RP/0/0/CPU0:SP1R3(config-vrf-export-rt)#int lo1
RP/0/0/CPU0:SP1R3(config-if)#vrf siteb
RP/0/0/CPU0:SP1R3(config-if)#ipv4 add 172.16.3.31/32
RP/0/0/CPU0:SP1R3(config-if)#router bgp 3356
RP/0/0/CPU0:SP1R3(config-bgp)# vrf siteb
RP/0/0/CPU0:SP1R3(config-bgp-vrf)#  rd 3:3
RP/0/0/CPU0:SP1R3(config-bgp-vrf)#  address-family ipv4 unicast
RP/0/0/CPU0:SP1R3(config-bgp-vrf-af)#    redistribute connected
```

How to speed up a configuration of R4? Copy the respective configuration from R3 before a commit by 'show config' and paste it to a notepad. Then edit it with the R4's settings. Next copy and paste the config from the notepad to R4.

```
RP/0/0/CPU0:SP1R3(config-bgp-vrf-af)#sh config
Building configuration...
!! IOS XR Configuration 6.1.2
vrf siteb
 address-family ipv4 unicast
  import route-target
   1:100
   3:3
  !
  export route-target
   3:3
  !
 !
!
interface Loopback1
 vrf siteb
 ipv4 address 172.16.3.31 255.255.255.255
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
router bgp 3356
 vrf siteb
  rd 3:3
  address-family ipv4 unicast
   redistribute connected
  !
 !
!
end
```

Verification. There are all prefixes advertised to RR.

```
SP1-R5-RR#sh bgp vpnv4 uni all
BGP table version is 130, local router ID is 33.56.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 3:3
 *>i  172.16.3.31/32   33.56.3.3                0    100      0 ?
Route Distinguisher: 4:4
 *>i  172.16.4.41/32   33.56.4.4                0    100      0 ?
Route Distinguisher: 111:100
 *>i  44.44.67.0/24    33.56.1.1                0    100      0 100 i
 *>i  172.16.6.61/32   33.56.1.1                0    100      0 100 i
 *>i  172.16.7.71/32   33.56.1.1                     100      0 100 i
Route Distinguisher: 112:100
 *>i  44.44.67.0/24    33.56.2.2                0    100      0 100 i
 *>i  172.16.6.61/32   33.56.2.2                     100      0 100 i
 *>i  172.16.7.71/32   33.56.2.2                0    100      0 100 i
```

But there are no newly advertised routes on R2 and R1.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 uni
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100
*>i44.44.67.0/24      33.56.1.1                0    100      0 100 i
*>i172.16.6.61/32     33.56.1.1                0    100      0 100 i
*>i172.16.7.71/32     33.56.1.1                     100      0 100 i
Route Distinguisher: 112:100 (default for vrf cust1)
* i44.44.67.0/24      33.56.1.1                0    100      0 100 i
*>                    40.60.27.7               0             0 100 i
* i172.16.6.61/32     33.56.1.1                0    100      0 100 i
*>                    40.60.27.7                             0 100 i
* i172.16.7.71/32     33.56.1.1                     100      0 100 i
*>                    40.60.27.7               0             0 100 i

Processed 6 prefixes, 9 paths
```

What is missing? Look at the 172.16.3.31/32 prefix.

```
SP1-R5-RR#sh bgp vpnv4 uni all 172.16.3.31
BGP routing table entry for 3:3:172.16.3.31/32, version 128
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     3
  Refresh Epoch 1
  Local, (Received from a RR-client)
    33.56.3.3 (metric 2) (via default) from 33.56.3.3 (33.56.3.3)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:3:3
      mpls labels in/out nolabel/103004
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      rx pathid: 0, tx pathid: 0x0
```

There is export RT 3:3. On R1 import RT is different.

```
RP/0/0/CPU0:SP1R2#sh run vrf cust1
vrf cust1
 address-family ipv4 unicast
  import route-target
   1:100
  !
  export route-target
   1:100
  !
 !
```

You need to match these attributes. Add RT import 3:3 on R2 and R1 to accept the prefixes.

```
RP/0/0/CPU0:SP1R2(config)#vrf cust1
RP/0/0/CPU0:SP1R2(config-vrf)# address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-vrf-af)#  import route-target
RP/0/0/CPU0:SP1R2(config-vrf-import-rt)#3:3
RP/0/0/CPU0:SP1R2(config-vrf-import-rt)#4:4
```

The prefixes are now imported.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 uni
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 3:3
*>i172.16.3.31/32     33.56.3.3                0    100      0 ?
Route Distinguisher: 4:4
*>i172.16.4.41/32     33.56.4.4                0    100      0 ?
Route Distinguisher: 111:100
*>i44.44.67.0/24      33.56.1.1                0    100      0 100 i
*>i172.16.6.61/32     33.56.1.1                0    100      0 100 i
*>i172.16.7.71/32     33.56.1.1                     100      0 100 i
Route Distinguisher: 112:100 (default for vrf cust1)
* i44.44.67.0/24      33.56.1.1                0    100      0 100 i
*>                    40.60.27.7               0             0 100 i
*>i172.16.3.31/32     33.56.3.3                0    100      0 ?
*>i172.16.4.41/32     33.56.4.4                0    100      0 ?
* i172.16.6.61/32     33.56.1.1                0    100      0 100 i
*>                    40.60.27.7                             0 100 i
* i172.16.7.71/32     33.56.1.1                     100      0 100 i
*>                    40.60.27.7               0             0 100 i

Processed 10 prefixes, 13 paths
```

Verify also prefixes from R1 and R2 on R3 and R4.

```
RP/0/0/CPU0:SP1R3#sh route vrf siteb
[..]

Gateway of last resort is not set

B    44.44.67.0/24 [200/0] via 33.56.1.1 (nexthop in vrf default), 00:10:51
L    172.16.3.31/32 is directly connected, 00:10:58, Loopback1
B    172.16.6.61/32 [200/0] via 33.56.1.1 (nexthop in vrf default), 00:10:51
B    172.16.7.71/32 [200/0] via 33.56.1.1 (nexthop in vrf default), 00:10:51
```

As you can see there is no 172.16.4.41/32 which is the expected result. R3's Lo1 and R4's Lo1 should not see each other.

Test the communication between R7 site A and site B/C.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CEL7#sh ip route 172.16.3.31
Routing entry for 172.16.3.31/32
  Known via "bgp 100", distance 20, metric 0
  Tag 3356, type external
  Last update from 40.60.27.2 00:05:02 ago
  Routing Descriptor Blocks:
  * 40.60.27.2, from 40.60.27.2, 00:05:02 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 3356
      MPLS label: none
```

There is the destination prefix in the RIB of CEL7. Verify the reachability..

```
CEL7#ping 172.16.3.31 sou 172.16.7.71
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.31, timeout is 2 seconds:
Packet sent with a source address of 172.16.7.71
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/11 ms

CEL7#trace 172.16.3.31 sou 172.16.7.71
Type escape sequence to abort.
Tracing the route to 172.16.3.31
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 3 msec 2 msec 1 msec
  2 33.56.23.3 10 msec *  11 msec

CEL7#trace 172.16.4.41 sou 172.16.7.71
Type escape sequence to abort.
Tracing the route to 172.16.4.41
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 2 msec 2 msec 2 msec
  2 33.56.24.4 12 msec *  12 msec
```

From the opposite direction.

```
RP/0/0/CPU0:SP1R3#sh cef vrf siteb 172.16.7.71
172.16.7.71/32, version 5, internal 0x5000001 0x0 (ptr 0xa18b79f4) [1], 0x0 (0x0), 0x208
(0xa1569208)
  Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 33.56.1.1/32, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa15d17f4 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 33.56.1.1/32 via 103003/0/21
     next hop 33.56.13.1/32 Gi0/0/0/6    labels imposed {0 101013}

RP/0/0/CPU0:SP1R3#trace vrf siteb 172.16.7.71

Type escape sequence to abort.
Tracing the route to 172.16.7.71

 1  33.56.13.1 [MPLS: Labels 0/101013 Exp 0] 19 msec  9 msec  0 msec
 2  41.61.16.6 19 msec  9 msec  0 msec
 3  44.44.67.7 9 msec  *  9 msec
```

Which of these labels {0 101013} is the VPN label?

```
RP/0/0/CPU0:SP1R3#sh bgp vpnv4 unicast labels
[..]
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop        Rcvd Label      Local Label
Route Distinguisher: 3:3 (default for vrf siteb)
*>i44.44.67.0/24      33.56.1.1       101004          nolabel
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
* i                    33.56.2.2       102004          nolabel
*> 172.16.3.31/32     0.0.0.0         nolabel         103004
*>i172.16.6.61/32     33.56.1.1       101012          nolabel
* i                    33.56.2.2       102011          nolabel
*>i172.16.7.71/32     33.56.1.1       101013          nolabel
* i                    33.56.2.2       102010          nolabel
Route Distinguisher: 111:100
*>i44.44.67.0/24      33.56.1.1       101004          nolabel
*>i172.16.6.61/32     33.56.1.1       101012          nolabel
*>i172.16.7.71/32     33.56.1.1       101013          nolabel
Route Distinguisher: 112:100
*>i44.44.67.0/24      33.56.2.2       102004          nolabel
*>i172.16.6.61/32     33.56.2.2       102011          nolabel
*>i172.16.7.71/32     33.56.2.2       102010          nolabel

Processed 10 prefixes, 13 paths
```

The label 101013 is the VPN label and is advertised by R1.

```
RP/0/0/CPU0:SP1R1#sh bgp vpnv4 unicast labels
[..]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop        Rcvd Label      Local Label
Route Distinguisher: 3:3
*>i172.16.3.31/32     33.56.3.3       103004          nolabel
Route Distinguisher: 4:4
*>i172.16.4.41/32     33.56.4.4       104004          nolabel
Route Distinguisher: 111:100 (default for vrf cust1)
* i44.44.67.0/24      33.56.2.2       102004          101004
*>                    41.61.16.6      nolabel         101004
*>i172.16.3.31/32     33.56.3.3       103004          nolabel
*>i172.16.4.41/32     33.56.4.4       104004          nolabel
* i172.16.6.61/32     33.56.2.2       102011          101012
*>                    41.61.16.6      nolabel         101012
* i172.16.7.71/32     33.56.2.2       102010          101013
*>                    41.61.16.6      nolabel         101013
Route Distinguisher: 112:100
*>i44.44.67.0/24      33.56.2.2       102004          nolabel
*>i172.16.6.61/32     33.56.2.2       102011          nolabel
*>i172.16.7.71/32     33.56.2.2       102010          nolabel
```

This is per prefix allocation mode. It requires higher label scalability from a device but it gives higher performance than other options.

You can allocate a label also per CE or per VRF.

```
RP/0/0/CPU0:SP1R1(config)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#vrf cust1
RP/0/0/CPU0:SP1R1(config-bgp-vrf)#add ipv4 unicast
RP/0/0/CPU0:SP1R1(config-bgp-vrf-af)#label mode ?
  per-ce          Set per CE label mode
  per-prefix      Set per perfix label mode
  per-vrf         Set per VRF label mode
  route-policy    Use a route policy to select prefixes for label allocation mode
```

Try the per VRF allocation mode.

```
RP/0/0/CPU0:SP1R1(config-bgp-vrf-af)#label mode per-vrf
```

You can see that for all prefixes in the VRF cust1 there is the same label 101005 allocated.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R1#sh bgp vpnv4 unicast labels
BGP router identifier 33.56.1.1, local AS number 3356
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 175
BGP NSR Initial initsync version 9 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop        Rcvd Label      Local Label
Route Distinguisher: 3:3
*>i172.16.3.31/32     33.56.3.3       103004          nolabel
Route Distinguisher: 4:4
*>i172.16.4.41/32     33.56.4.4       104004          nolabel
Route Distinguisher: 111:100 (default for vrf cust1)
* i44.44.67.0/24      33.56.2.2       102004          101005
*>                    41.61.16.6      nolabel         101005
*>i172.16.3.31/32     33.56.3.3       103004          nolabel
*>i172.16.4.41/32     33.56.4.4       104004          nolabel
* i172.16.6.61/32     33.56.2.2       102011          101005
*>                    41.61.16.6      nolabel         101005
* i172.16.7.71/32     33.56.2.2       102010          101005
*>                    41.61.16.6      nolabel         101005
Route Distinguisher: 112:100
*>i44.44.67.0/24      33.56.2.2       102004          nolabel
*>i172.16.6.61/32     33.56.2.2       102011          nolabel
*>i172.16.7.71/32     33.56.2.2       102010          nolabel

Processed 10 prefixes, 13 paths
```

There is just one VPN label for all prefixes in VRF cust1 advertised by R1.

```
RP/0/0/CPU0:SP1R1#sh mpls for
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
101000 Exp-Null-v4 33.56.2.2/32       Gi0/0/0/2    33.56.12.2      1750189
101001 Exp-Null-v4 33.56.3.3/32       Gi0/0/0/5    33.56.13.3      1738072
101002 Exp-Null-v4 33.56.4.4/32       Gi0/0/0/0    33.56.14.4      1739426
101003 Exp-Null-v4 33.56.5.5/32       Gi0/0/0/3    33.56.15.5      28487559
101005 Aggregate   cust1: Per-VRF Aggr[V]   \
                                      cust1                        756
101007 Unlabelled  2001:44:44:67::/64[V]   \
                                      Gi0/0/0/1    fe80::6         0
```

Compare this table with R2. There are 3 labels instead of 1.

```
RP/0/0/CPU0:SP1R2#sh mpls for
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
102000 Exp-Null-v4 33.56.4.4/32       Gi0/0/0/4    33.56.24.4      1564384
102001 Exp-Null-v4 33.56.3.3/32       Gi0/0/0/0    33.56.23.3      1738632
102002 103000      33.56.5.5/32       Gi0/0/0/0    33.56.23.3      525
       101003      33.56.5.5/32       Gi0/0/0/2    33.56.12.1      2339046
102003 Exp-Null-v4 33.56.1.1/32       Gi0/0/0/2    33.56.12.1      1749132
102004 Unlabelled  44.44.67.0/24[V]   Gi0/0/0/1    40.60.27.7      0
102007 Unlabelled  2001:44:44:67::/64[V]   \
                                      Gi0/0/0/1    fe80::7         0
102010 Unlabelled  172.16.7.71/32[V]  Gi0/0/0/1    40.60.27.7      2528
102011 Unlabelled  172.16.6.61/32[V]  Gi0/0/0/1    40.60.27.7      0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The other label in {0 101013} of the CEF output of SP1R3 is the label 0 or Exp-Null. This is the transport label.

```
RP/0/0/CPU0:SP1R3#sh mpls forwarding prefix 33.56.1.1/32
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
103003 Exp-Null-v4 33.56.1.1/32       Gi0/0/0/6    33.56.13.1      1736716
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.6: Shared services

You will configure a shared VRF which simulates centralized resources or the exit to the Internet. Add Loopback1 on R2 with the following settings.

R2:

IP address of Loopback1: 172.16.2.21/32

VRF name: shared

RD: 111:100

RT: 2:2

```
RP/0/0/CPU0:SP1R2(config)#vrf shared
RP/0/0/CPU0:SP1R2(config-vrf)# address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-vrf-af)#  import route-target
RP/0/0/CPU0:SP1R2(config-vrf-import-rt)#   2:2
RP/0/0/CPU0:SP1R2(config-vrf-import-rt)#  export route-target
RP/0/0/CPU0:SP1R2(config-vrf-export-rt)#   2:2
RP/0/0/CPU0:SP1R2(config-vrf-export-rt)#int lo1
RP/0/0/CPU0:SP1R2(config-if)#vrf shared
RP/0/0/CPU0:SP1R2(config-if)#ipv4 add 172.16.2.21/32
RP/0/0/CPU0:SP1R2(config-if)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)# vrf shared
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#  rd 111:100
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#  address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-vrf-af)#  redistribute connected
```

There is no prefix 172.16.2.21/32 on R1 even though RDs 111:100 are the same.

```
RP/0/0/CPU0:SP1R1#sh bgp vpnv4 unicast
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 3:3
*>i172.16.3.31/32     33.56.3.3                0    100      0 ?
Route Distinguisher: 4:4
*>i172.16.4.41/32     33.56.4.4                0    100      0 ?
Route Distinguisher: 111:100 (default for vrf cust1)
* i44.44.67.0/24      33.56.2.2                0    100      0 100 i
*>                    41.61.16.6               0             0 100 i
*>i172.16.3.31/32     33.56.3.3                0    100      0 ?
*>i172.16.4.41/32     33.56.4.4                0    100      0 ?
* i172.16.6.61/32     33.56.2.2                     100      0 100 i
*>                    41.61.16.6               0             0 100 i
* i172.16.7.71/32     33.56.2.2                0    100      0 100 i
*>                    41.61.16.6                             0 100 i
Route Distinguisher: 112:100
*>i44.44.67.0/24      33.56.2.2                0    100      0 100 i
*>i172.16.6.61/32     33.56.2.2                     100      0 100 i
*>i172.16.7.71/32     33.56.2.2                0    100      0 100 i

Processed 10 prefixes, 13 paths
```

To import this prefix to all necessary VRFs a proper RT should be set. RD does not matter. Let's make the VRF named 'shared' a real shared resource so it can be accessible from every VRF. Configure correct RTs in one place in the VRF 'shared'.

Verify your configuration.

```
RP/0/0/CPU0:SP1R1#sh bgp vpnv4 unicast rd 111:100 172.16.2.21/32
BGP routing table entry for 172.16.2.21/32, Route Distinguisher: 111:100
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker               187         187
Last Modified: Jan 01 22:55:27.449 for 00:02:42
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    33.56.2.2 (metric 2) from 33.56.5.5 (33.56.2.2)
      Received Label 102005
      Origin incomplete, metric 0, localpref 100, valid, internal, best, group-best,
import-candidate, imported
      Received Path ID 0, Local Path ID 0, version 187
      Extended community: RT:1:100 RT:2:2 RT:3:3 RT:4:4
      Originator: 33.56.2.2, Cluster list: 33.56.5.5
      Source AFI: VPNv4 Unicast, Source VRF: cust1, Source Route Distinguisher: 111:100
```

This prefix can be in more than just one VPNs right now.

```
RP/0/0/CPU0:SP1R3#sh bgp vpnv4 uni 172.16.2.21/32 longer-prefixes
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 3:3 (default for vrf siteb)
*>i172.16.2.21/32     33.56.2.2                0    100      0 ?
Route Distinguisher: 111:100
*>i172.16.2.21/32     33.56.2.2                0    100      0 ?

Processed 2 prefixes, 2 paths
```

Check your RT's settings:

```
RP/0/0/CPU0:SP1R2#sh vrf shared ipv4 det
VRF shared; RD 111:100; VPN ID not set
VRF mode: Regular
Description not set
Interfaces:
  Loopback1
Address family IPV4 Unicast
  Import VPN route-target communities:
    RT:2:2
    RT:3:3
    RT:4:4
    RT:1:100
  Export VPN route-target communities:
    RT:2:2
    RT:3:3
    RT:4:4
    RT:1:100
  No import route policy
  No export route policy
```

Verify the RIB table. Now there is a complete set of prefixes.

```
RP/0/0/CPU0:SP1R2#sh route vrf shared
[..]

Gateway of last resort is not set

B    44.44.67.0/24 [20/0] via 40.60.27.7 (nexthop in vrf cust1), 00:10:23
L    172.16.2.21/32 is directly connected, 00:21:01, Loopback1
B    172.16.3.31/32 [200/0] via 33.56.3.3 (nexthop in vrf default), 00:10:23
B    172.16.4.41/32 [200/0] via 33.56.4.4 (nexthop in vrf default), 00:10:23
B    172.16.6.61/32 [20/0] via 40.60.27.7 (nexthop in vrf cust1), 00:10:23
B    172.16.7.71/32 [20/0] via 40.60.27.7 (nexthop in vrf cust1), 00:10:23
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Verify reachability.

```
RP/0/0/CPU0:SP1R3#ping vrf siteb 172.16.2.21
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.21, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/9 ms

RP/0/0/CPU0:SP1R4#ping vrf sitec 172.16.2.21
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.21, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/37/79 ms

CEL7#ping 172.16.2.21 source 172.16.7.71
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.21, timeout is 2 seconds:
Packet sent with a source address of 172.16.7.71
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms

CEL6#ping 172.16.2.21 source 172.16.6.61
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.21, timeout is 2 seconds:
Packet sent with a source address of 172.16.6.61
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/13/44 ms
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: MPLS transport for IPv6

Task 2.1: 6VPE Service

The task is to configure IPv6 L3VPN services without using IPv6 addressing in the core network. Bring back IPv6 eBGP settings on R6 (CEL6) and R7 (CEL7) which could be disabled in Task 1.

Move IPv6 eBGP sessions CEL7-R2 and CEL6-R1 to VRF cust1 similarly as it was done for IPv4.

A verification step. On RR the VPNv6 sessions should be up with R1 and R2.

```
SP1-R5-RR#sh bgp vpnv6 unicast all summ
BGP router identifier 33.56.5.5, local AS number 3356
BGP table version is 1, main routing table version 1
6 network entries using 1680 bytes of memory
6 path entries using 936 bytes of memory
2/0 BGP path/bestpath attribute entries using 576 bytes of memory
3 BGP AS-PATH entries using 88 bytes of memory
1 BGP extended community entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3304 total bytes of memory
BGP activity 1538/1516 prefixes, 1638/1602 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
33.56.1.1       4         3356       9       12        1    0    0 00:00:30        3
33.56.2.2       4         3356       9       12        1    0    0 00:00:25        3
33.56.3.3       4         3356       0        0        1    0    0 never    (NoNeg)
33.56.4.4       4         3356       0        0        1    0    0 never    (NoNeg)

SP1-R5-RR#sh bgp vpnv6 unicast all
BGP table version is 7, local router ID is 33.56.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100
 *>i  2001:44:44:67::/64
                      ::FFFF:33.56.1.1
                                               0    100      0 100 i
 *>i  2001:44:66:CC::6/128
                      ::FFFF:33.56.1.1
                                               0    100      0 100 i
 *>i  2001:44:66:CC::7/128
                      ::FFFF:33.56.1.1
                                                    100      0 100 i
Route Distinguisher: 112:100
 *>i  2001:44:44:67::/64
                      ::FFFF:33.56.2.2
     Network          Next Hop            Metric LocPrf Weight Path
                                               0    100      0 100 i
 *>i  2001:44:66:CC::6/128
                      ::FFFF:33.56.2.2
                                                    100      0 100 i
 *>i  2001:44:66:CC::7/128
                      ::FFFF:33.56.2.2
                                               0    100      0 100 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

You can see that the next hop is composed by prepending the value of ::FFFF: to an IPv4 address used to establish a BGP session. This next hop is the IPv4-mapped IPv6 address.

```
RP/0/0/CPU0:SP1R1#sh bgp vpnv6 unicast
[..]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100 (default for vrf cust1)
* i2001:44:44:67::/64 33.56.2.2                0    100      0 100 i
*>                    2001:41:61:16::6
                                             0             0 100 i
* i2001:44:66:cc::6/128
                      33.56.2.2                    100      0 100 i
*>                    2001:41:61:16::6
                                             0             0 100 i
* i2001:44:66:cc::7/128
                      33.56.2.2              0    100      0 100 i
*>                    2001:41:61:16::6
                                                           0 100 i
Route Distinguisher: 112:100
*>i2001:44:44:67::/64 33.56.2.2              0    100      0 100 i
*>i2001:44:66:cc::6/128
                      33.56.2.2                    100      0 100 i
*>i2001:44:66:cc::7/128
                      33.56.2.2              0    100      0 100 i

Processed 6 prefixes, 9 paths


RP/0/0/CPU0:SP1R2#sh bgp vpnv6 unicast
[..]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100
*>i2001:44:44:67::/64 33.56.1.1              0    100      0 100 i
*>i2001:44:66:cc::6/128
                      33.56.1.1              0    100      0 100 i
*>i2001:44:66:cc::7/128
                      33.56.1.1                    100      0 100 i
Route Distinguisher: 112:100 (default for vrf cust1)
* i2001:44:44:67::/64 33.56.1.1                0    100      0 100 i
*>                    2001:40:60:27::7
                                             0             0 100 i
* i2001:44:66:cc::6/128
                      33.56.1.1              0    100      0 100 i
*>                    2001:40:60:27::7
                                                           0 100 i
* i2001:44:66:cc::7/128
                      33.56.1.1                    100      0 100 i
*>                    2001:40:60:27::7
                                             0             0 100 i

Processed 6 prefixes, 9 paths
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Provide communication between VRFs. Configure IPv6 addresses in VRFs: siteb on R3 and sitec on R4.

```
RP/0/0/CPU0:SP1R3(config)#int lo1
RP/0/0/CPU0:SP1R3(config-if)#ipv6 add 2001:33:56::3:31/128
RP/0/0/CPU0:SP1R3(config-if)#router bgp 3356
RP/0/0/CPU0:SP1R3(config-bgp)#add vpnv6 uni
RP/0/0/CPU0:SP1R3(config-bgp)#neighbor 33.56.5.5
RP/0/0/CPU0:SP1R3(config-bgp-nbr)#add vpnv6 uni
RP/0/0/CPU0:SP1R3(config-bgp-af)#vrf siteb
RP/0/0/CPU0:SP1R3(config-bgp-vrf)#add ipv6 uni
RP/0/0/CPU0:SP1R3(config-bgp-vrf-af)#redistribute connected

RP/0/0/CPU0:SP1R4(config)#interface lo1
RP/0/0/CPU0:SP1R4(config-if)# ipv6 add 2001:33:56::4:41/128
RP/0/0/CPU0:SP1R4(config-if)#router bgp 3356
RP/0/0/CPU0:SP1R4(config-bgp)#add vpnv6 uni
RP/0/0/CPU0:SP1R3(config-bgp)#neighbor 33.56.5.5
RP/0/0/CPU0:SP1R3(config-bgp-nbr)#add vpnv6 uni
RP/0/0/CPU0:SP1R4(config-bgp-af)#vrf sitec
RP/0/0/CPU0:SP1R4(config-bgp-vrf)#add ipv6 uni
RP/0/0/CPU0:SP1R4(config-bgp-vrf-af)#redistribute connected
```

Set the correct RT values by yourself on R1-R4 to import prefixes.

Verification. On RR the VPNv6 sessions should be up with R3 and R4.

```
SP1-R5-RR#sh bgp vpnv6 unicast all summ
[..]

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
33.56.1.1       4         3356   11544   12765       32    0    0 1w1d              3
33.56.2.2       4         3356   11550   12791       32    0    0 1w1d              3
33.56.3.3       4         3356      24      42       32    0    0 00:14:08          1
33.56.4.4       4         3356      22      36       32    0    0 00:12:33          1
```

CEL7 and CEL6 have IPv6 routes received from R3 and R4.

```
CEL7#sh ipv6 route
[..]
C   2001:22:42:37::/64 [0/0]
     via GigabitEthernet4, directly connected
L   2001:22:42:37::7/128 [0/0]
     via GigabitEthernet4, receive
B   2001:33:56::3:31/128 [20/0]
     via FE80::2, GigabitEthernet2
B   2001:33:56::4:41/128 [20/0]
     via FE80::2, GigabitEthernet2
C   2001:40:60:27::/64 [0/0]
     via GigabitEthernet2, directly connected
L   2001:40:60:27::7/128 [0/0]
     via GigabitEthernet2, receive
C   2001:44:44:67::/64 [0/0]
     via GigabitEthernet3, directly connected
L   2001:44:44:67::7/128 [0/0]
     via GigabitEthernet3, receive
O   2001:44:66:CC::6/128 [110/1]
     via FE80::6, GigabitEthernet3
LC  2001:44:66:CC::7/128 [0/0]
     via Loopback0, receive
L   FF00::/8 [0/0]
     via Null0, receive
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Check the prefix in the FIB table along with the VPN and transport label.

```
RP/0/0/CPU0:SP1R2#sh cef vrf cust1 ipv6 2001:33:56::3:31/128
2001:33:56::3:31/128, version 50, internal 0x5000001 0x0 (ptr 0xa11bf374) [1], 0x0
(0x0), 0x208 (0xa15692a8)
 Updated Jan 01 00:25:51.331
 Prefix Len 128, traffic index 0, precedence n/a, priority 3
   via ::ffff:33.56.3.3/128, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa194b0cc 0x0]
    recursion-via-/128
    next hop VRF - 'default', table - 0xe0000000
    next hop ::ffff:33.56.3.3/128 via 102001/0/21
     next hop 33.56.23.3/32 Gi0/0/0/0     labels imposed {0 103005}
```

The label 103005 is the VPN label assigned by the BGP process on the egress PE/R3.

```
RP/0/0/CPU0:SP1R3#sh bgp vpnv6 unicast rd 3:3 2001:33:56::3:31/128
BGP routing table entry for 2001:33:56::3:31/128, Route Distinguisher: 3:3
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 10          10
    Local Label: 103005
Last Modified: Jan 01 00:22:50.115 for 00:14:06
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    33.56.5.5
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    33.56.5.5
  Local
    :: from 0.0.0.0 (33.56.3.3)
      Origin incomplete, metric 0, localpref 100, weight 32768, valid, redistributed,
best, group-best, import-candidate
      Received Path ID 0, Local Path ID 0, version 10
      Extended community: RT:1:100
```

The same verification step on SP1R2 for the prefix 2001:33:56::4:41/128.

```
RP/0/0/CPU0:SP1R2#sh cef vrf cust1 ipv6 2001:33:56::4:41/128
2001:33:56::4:41/128, version 51, internal 0x5000001 0x0 (ptr 0xa11bf3f4) [1], 0x0
(0x0), 0x208 (0xa1569320)
 Updated Jan 01 00:27:25.954
 Prefix Len 128, traffic index 0, precedence n/a, priority 3
   via ::ffff:33.56.4.4/128, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa194b148 0x0]
    recursion-via-/128
    next hop VRF - 'default', table - 0xe0000000
    next hop ::ffff:33.56.4.4/128 via 102000/0/21
     next hop 33.56.24.4/32 Gi0/0/0/4     labels imposed {0 104005}


RP/0/0/CPU0:SP1R4#sh bgp vpnv6 unicast rd 4:4 2001:33:56::4:41/128
BGP routing table entry for 2001:33:56::4:41/128, Route Distinguisher: 4:4
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                  4           4
    Local Label: 104005
Last Modified: Jan 01 00:25:35.209 for 00:13:02
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    33.56.5.5
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    33.56.5.5
  Local
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    :: from 0.0.0.0 (33.56.4.4)
      Origin incomplete, metric 0, localpref 100, weight 32768, valid, redistributed,
best, group-best, import-candidate
      Received Path ID 0, Local Path ID 0, version 4
      Extended community: RT:1:100
```

Verify the data plane.

```
CEL7#ping 2001:33:56::3:31 sou 2001:44:66:CC::7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:33:56::3:31, timeout is 2 seconds:
Packet sent with a source address of 2001:44:66:CC::7
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 7/8/10 ms

CEL7#ping 2001:33:56::4:41 sou 2001:44:66:CC::7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:33:56::4:41, timeout is 2 seconds:
Packet sent with a source address of 2001:44:66:CC::7
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/10/13 ms

RP/0/0/CPU0:SP1R3#ping vrf siteb 2001:33:56::4:41
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:33:56::4:41, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/9 ms
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.2: 6PE Service

The task is to configure IPv6 transport services without using IPv6 addressing in the core network.

Check the current IPv6 native communication.

The BGP table.

```
RP/0/0/CPU0:SP1R2#sh bgp ipv6 unicast 2001:22:44:ee::6
BGP routing table entry for 2001:22:44:ee::6/128
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 37          37
Last Modified: Jan 01 0 00:35:03.950 for 1w1d
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  1239 200
    2001:33:56:aa::3 (metric 1) from 2001:33:56:aa::5 (33.56.3.3)
      Origin IGP, localpref 100, valid, internal, best, group-best
      Received Path ID 0, Local Path ID 0, version 37
      Originator: 33.56.3.3, Cluster list: 33.56.5.5
```

The FIB table for the same IP address.

```
RP/0/0/CPU0:SP1R2#sh cef ipv6 2001:22:44:ee::6
2001:22:44:ee::6/128, version 90, internal 0x5000001 0x0 (ptr 0xa11beb74) [1], 0x0
(0x0), 0x0 (0x0)
 Updated Jan 01 00:35:03.503
 local adjacency fe80::3
 Prefix Len 128, traffic index 0, precedence n/a, priority 4
   via 2001:33:56:aa::3/128, 2 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa11be574 0x0]
    next hop 2001:33:56:aa::3/128 via 2001:33:56:aa::3/128
```

Reachability verification. The hop by hop routing is performed based on the IPv6 destination address.

```
RP/0/0/CPU0:SP1R2#trace 2001:22:44:ee::6 sou 2001:33:56:aa::2

Type escape sequence to abort.
Tracing the route to 2001:22:44:ee::6

 1  2001:33:56:23::3 29 msec 19 msec 0 msec
 2  2001:22:22:33::254 9 msec 9 msec 9 msec
 3  2001:12:39:13::1 9 msec 9 msec 9 msec
 4  2001:21:41:16::6 19 msec 19 msec 19 msec
```

Configure IPv6 labelled transport service. Turn on the IPv6 labelled AFI. Repeat the following configuration on R1, R2, R4.

```
RP/0/0/CPU0:SP1R3(config)#router bgp 3356
RP/0/0/CPU0:SP1R3(config-bgp)#add ipv6 uni
RP/0/0/CPU0:SP1R3(config-bgp-af)#allocate-label all
RP/0/0/CPU0:SP1R3(config-bgp-af)#nei 33.56.5.5
RP/0/0/CPU0:SP1R3(config-bgp-nbr)#add ipv6 labelled-unicast
RP/0/0/CPU0:SP1R3(config-bgp-nbr-af)#next-hop-self
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Configure send-label capability on RR to advertise labels to PEs.

```
SP1-R5-RR(config)#router bgp 3356
SP1-R5-RR(config-router)#add ipv6 uni
SP1-R5-RR(config-router-af)#neighbor 33.56.1.1 activate
SP1-R5-RR(config-router-af)#neighbor 33.56.1.1 send-label
SP1-R5-RR(config-router-af)#neighbor 33.56.1.1 route-reflector-client
SP1-R5-RR(config-router-af)#neighbor 33.56.2.2 activate
SP1-R5-RR(config-router-af)#neighbor 33.56.2.2 send-label
SP1-R5-RR(config-router-af)#neighbor 33.56.2.2 route-reflector-client
SP1-R5-RR(config-router-af)#neighbor 33.56.3.3 activate
SP1-R5-RR(config-router-af)#neighbor 33.56.3.3 send-label
SP1-R5-RR(config-router-af)#neighbor 33.56.3.3 route-reflector-client
SP1-R5-RR(config-router-af)#neighbor 33.56.4.4 activate
SP1-R5-RR(config-router-af)#neighbor 33.56.4.4 send-label
SP1-R5-RR(config-router-af)#neighbor 33.56.4.4 route-reflector-client
SP1-R5-RR(config-router-af)#no neighbor 2001:33:56:AA::1 activate
SP1-R5-RR(config-router-af)#no neighbor 2001:33:56:AA::2 activate
SP1-R5-RR(config-router-af)#no neighbor 2001:33:56:AA::3 activate
SP1-R5-RR(config-router-af)#no neighbor 2001:33:56:AA::4 activate
```

Verify assigned labels.

```
RP/0/0/CPU0:SP1R2#sh bgp ipv6 unicast labels
[..]
   Network            Next Hop        Rcvd Label      Local Label
*>i2001:12:39::/48    33.56.3.3       103006          nolabel
*>i2001:22:44:ee::6/128
                      33.56.3.3       103007          nolabel
*>i2001:22:44:ee::7/128
                      33.56.3.3       103008          nolabel
* i2001:33:56::/48    33.56.1.1       101011          102014
*>                    ::              nolabel         102014
s> 2001:33:56:12::/64 ::              nolabel         102014
s> 2001:33:56:23::/64 ::              nolabel         102014
s> 2001:33:56:24::/64 ::              nolabel         102014
*>i2001:66:66:67::/64 33.56.3.3       103010          nolabel

Processed 8 prefixes, 9 paths
```

Check the BGP global table.

```
RP/0/0/CPU0:SP1R2#sh bgp ipv6 unicast 2001:22:44:ee::6
BGP routing table entry for 2001:22:44:ee::6/128
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 53          53
Last Modified: Jan 01 01:23:41.950 for 00:00:55
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  1239 200
    33.56.3.3 (metric 2) from 33.56.5.5 (33.56.3.3)
      Received Label 103007
      Origin IGP, localpref 100, valid, internal, best, group-best
      Received Path ID 0, Local Path ID 0, version 53
      Originator: 33.56.3.3, Cluster list: 33.56.5.5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The FIB table:

```
RP/0/0/CPU0:SP1R2#sh cef ipv6 2001:22:44:ee::6
2001:22:44:ee::6/128, version 114, internal 0x5000001 0x0 (ptr 0xa11bf574) [1], 0x0
(0x0), 0x208 (0xa1569258)
 Updated Jan 01 01:23:41.593
 Prefix Len 128, traffic index 0, precedence n/a, priority 4
   via ::ffff:33.56.3.3/128, 5 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa194b0cc 0x0]
    recursion-via-/128
    next hop VRF - 'default', table - 0xe0000000
    next hop ::ffff:33.56.3.3/128 via 102001/0/21
     next hop 33.56.23.3/32 Gi0/0/0/0     labels imposed {0 103007}
```

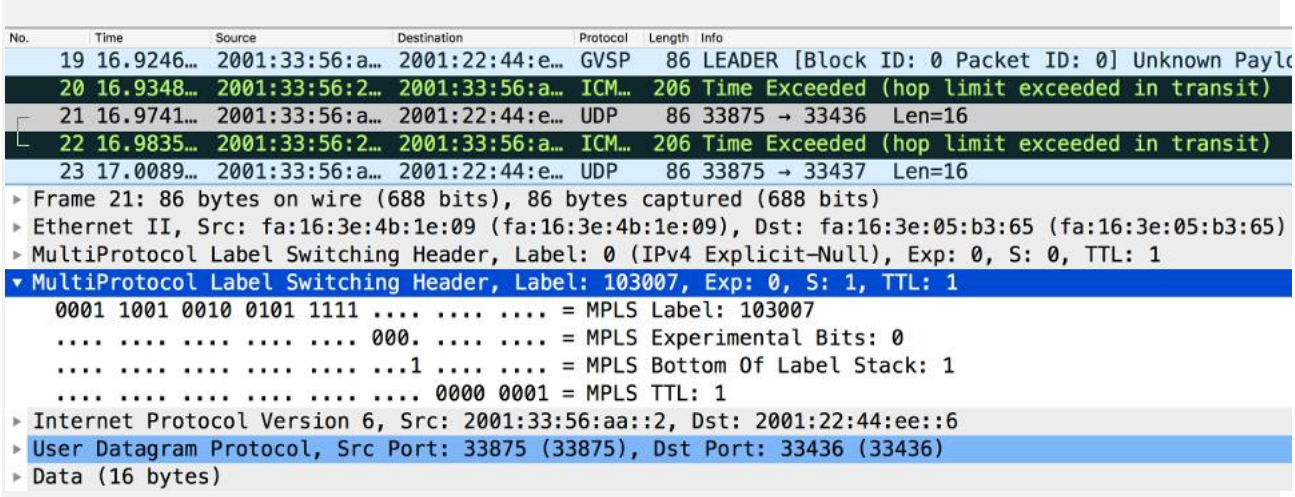
Reachability verification.

```
RP/0/0/CPU0:SP1R2#trace 2001:22:44:ee::6 sou 2001:33:56:aa::2

Type escape sequence to abort.
Tracing the route to 2001:22:44:ee::6

 1  2001:33:56:23::3 [MPLS: Labels 0/103007 Exp 0] 19 msec 9 msec 9 msec
 2  2001:22:22:33::254 9 msec 9 msec 9 msec
 3  2001:12:39:13::1 9 msec 9 msec 9 msec
 4  2001:21:41:16::6 19 msec 19 msec 29 msec
```

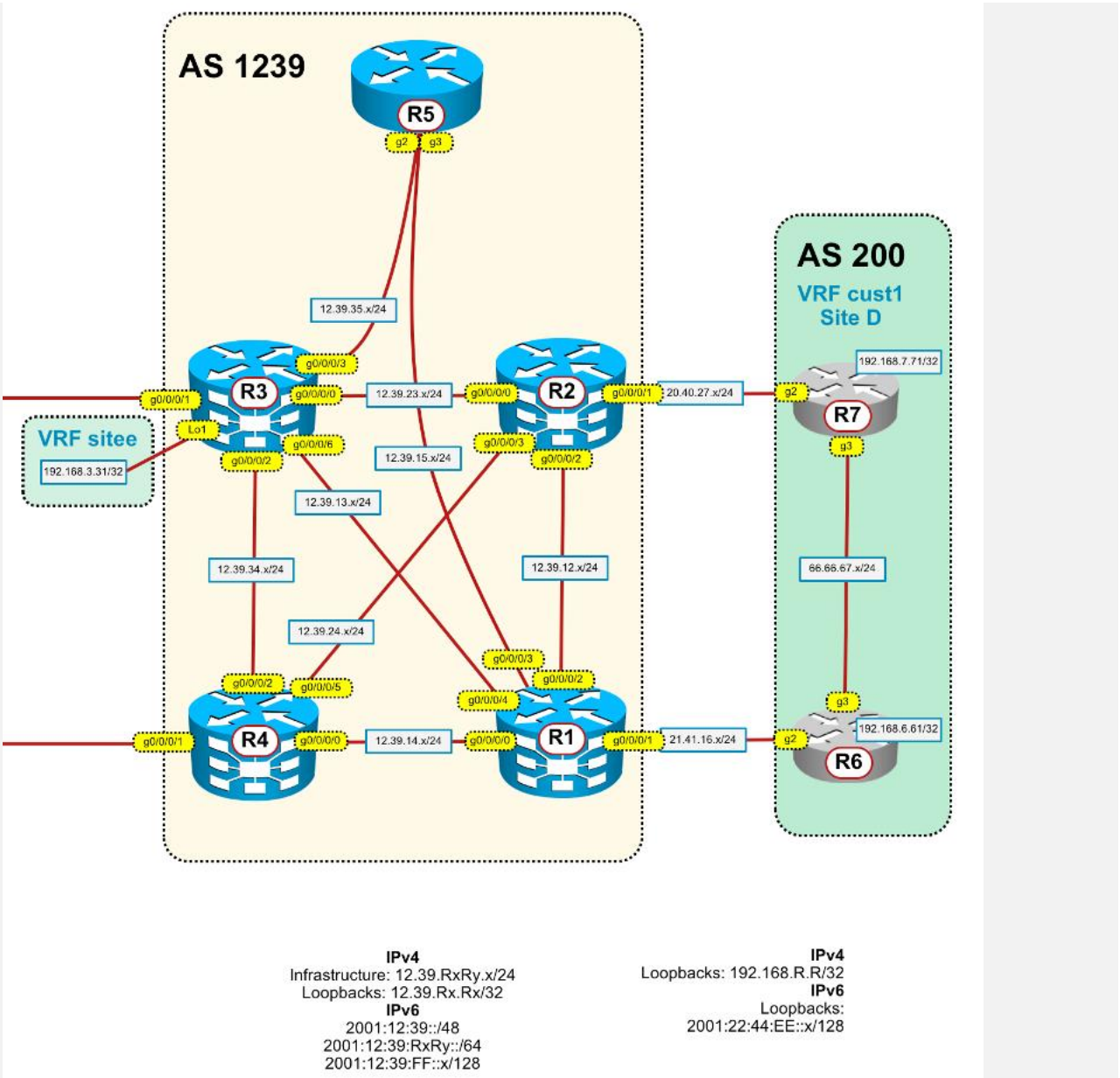
The next picture is a packet capture from R2 Gi0/0/0/0. You can see two labels. The transport label: 0 and the VPN label: 103007. This time a forwarding decision is made based on the label not on the IPv6 address. Labels are removed on R3 as the rest of the path is unlabelled based on the IPv6 destination address.



© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3: L3VPN in AS 1239

A customer named cust1 is also located in AS 200 Site D. Connect this site to the L3VPN service offered by the Service Provider AS 1239. Prepare the necessary configurations.



© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.1: Enable L3VPN

Enable the VPNv4 AFI between iBGP peers in AS 1239.

Task 3.2: Configure Site D prefixes

Configure Loopback1 on R6 and R7 with the following settings.

R6:

IP address of Loopback1: 192.168.6.61/32

R7:

IP address of Loopback1: 192.168.7.71/32

Task 3.3: Connect VRF cust1 Site D

Connect Site D to the VRF cust1 using the following attributes.

R1:

VRF name: cust1

RD: 200:1

RT: 2:2

R2:

VRF name: cust1

RD: 200:2

RT: 2:2

Task 3.4: Connect VRF sitee

Connect Site E to the L3VPN service. Use the following settings.

R3:

VRF name: sitee

RD: 200:3

RT: 3:3

IP address of Loopback1: 192.168.3.31/32

Advertise all 3 prefixes related to Loopback1 interfaces from Site D and E. Add necessary route-target values to achieve full communication between sites.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.5: Verify prefix advertisement

Verify the prefix advertisement and connectivity.

```
SP2-R5-RR#sh bgp vpnv4 uni all summ
BGP router identifier 12.39.5.5, local AS number 1239
[..]

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
12.39.1.1       4         1239      22      35       14    0    0 00:16:27        3
12.39.2.2       4         1239      24      39       14    0    0 00:18:46        3
12.39.3.3       4         1239      23      35       14    0    0 00:15:52        1
12.39.4.4       4         1239      21      28       14    0    0 00:15:42        0

SP2-R5-RR#sh bgp vpnv4 uni all
[..]

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1
 *>i  66.66.67.0/24    12.39.2.2                0    100      0 200 i
 *>i  192.168.6.61/32  12.39.2.2                     100      0 200 i
 *>i  192.168.7.71/32  12.39.2.2                0    100      0 200 i
Route Distinguisher: 200:2
 *>i  66.66.67.0/24    12.39.1.1                0    100      0 200 i
 *>i  192.168.6.61/32  12.39.1.1                0    100      0 200 i
 *>i  192.168.7.71/32  12.39.1.1                     100      0 200 i
Route Distinguisher: 200:3
 *>i  192.168.3.31/32  12.39.3.3                0    100      0 ?


RP/0/0/CPU0:SP2R2#sh bgp vpnv4 uni
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1 (default for vrf cust1)
* i66.66.67.0/24      12.39.1.1                0    100      0 200 i
*>                    20.40.27.7               0             0 200 i
*>i192.168.3.31/32    12.39.3.3                0    100      0 ?
* i192.168.6.61/32    12.39.1.1                0    100      0 200 i
*>                    20.40.27.7                             0 200 i
* i192.168.7.71/32    12.39.1.1                     100      0 200 i
*>                    20.40.27.7               0             0 200 i
Route Distinguisher: 200:2
*>i66.66.67.0/24      12.39.1.1                0    100      0 200 i
*>i192.168.6.61/32    12.39.1.1                0    100      0 200 i
*>i192.168.7.71/32    12.39.1.1                     100      0 200 i
Route Distinguisher: 200:3
*>i192.168.3.31/32    12.39.3.3                0    100      0 ?

Processed 8 prefixes, 11 paths

RP/0/0/CPU0:SP2R1#sh bgp vpnv4 uni
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1
*>i66.66.67.0/24      12.39.2.2                0    100      0 200 i
*>i192.168.6.61/32    12.39.2.2                     100      0 200 i
*>i192.168.7.71/32    12.39.2.2                0    100      0 200 i
Route Distinguisher: 200:2 (default for vrf cust1)
* i66.66.67.0/24      12.39.2.2                0    100      0 200 i
*>                    21.41.16.6               0             0 200 i
*>i192.168.3.31/32    12.39.3.3                0    100      0 ?
* i192.168.6.61/32    12.39.2.2                     100      0 200 i
*>                    21.41.16.6               0             0 200 i
* i192.168.7.71/32    12.39.2.2                0    100      0 200 i
*>                    21.41.16.6                             0 200 i
Route Distinguisher: 200:3
*>i192.168.3.31/32    12.39.3.3                0    100      0 ?
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Processed 8 prefixes, 11 paths

CER7#sh ip route bgp
[..]

      192.168.3.0/32 is subnetted, 1 subnets
B        192.168.3.31 [20/0] via 20.40.27.2, 00:01:27
      192.168.6.0/32 is subnetted, 2 subnets
B        192.168.6.61 [200/0] via 192.168.6.6, 00:24:32


CER6#sh ip route bgp
[..]

      192.168.3.0/32 is subnetted, 1 subnets
B        192.168.3.31 [20/0] via 21.41.16.1, 00:00:58
      192.168.7.0/32 is subnetted, 2 subnets
B        192.168.7.71 [200/0] via 192.168.7.7, 00:24:02


RP/0/0/CPU0:SP2R3#sh cef vrf sitee 192.168.6.61
192.168.6.61/32, version 4, internal 0x5000001 0x0 (ptr 0xa1899b74) [1], 0x0 (0x0),
0x208 (0xa15690f0)
 Updated Jan 01 18:30:17.365
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 12.39.1.1/32, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa15d17f4 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 12.39.1.1/32 via 123302/0/21
     next hop 12.39.13.1/32 Gi0/0/0/6    labels imposed {0 123105}
```

Reachability verification.

```
CER6#trace 192.168.3.31 sou 192.168.6.61
Type escape sequence to abort.
Tracing the route to 192.168.3.31
VRF info: (vrf in name/id, vrf out name/id)
  1 21.41.16.1 4 msec 1 msec 2 msec
  2 12.39.13.3 12 msec *  10 msec
```

No.	Time	Source	Destination	Protocol	Length	Info
15	13.6111…	12.39.13.1	224.0.0.2	LDP	84	Hello Message
16	14.2806…	12.39.13.3	224.0.0.2	LDP	84	Hello Message
17	15.8172…	192.168.6.61	192.168.3.31	ICMP	122	Echo (ping) request id=0x0000, seq=0/0, ttl=254 (reply…
18	15.8206…	192.168.3.31	192.168.6.61	ICMP	122	Echo (ping) reply id=0x0000, seq=0/0, ttl=255 (reque…
19	15.8268…	192.168.6.61	192.168.3.31	ICMP	122	Echo (ping) request id=0x0000, seq=1/256, ttl=254 (rep…

```
▸ Frame 18: 122 bytes on wire (976 bits), 122 bytes captured (976 bits)
▸ Ethernet II, Src: fa:16:3e:21:f7:5e (fa:16:3e:21:f7:5e), Dst: fa:16:3e:52:d1:35 (fa:16:3e:52:d1:35)
▾ MultiProtocol Label Switching Header, Label: 0 (IPv4 Explicit-Null), Exp: 0, S: 0, TTL: 255
    0000 0000 0000 0000 0000 .... .... .... = MPLS Label: IPv4 Explicit-Null (0)
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...0 .... .... = MPLS Bottom Of Label Stack: 0
    .... .... .... .... .... .... 1111 1111 = MPLS TTL: 255
▾ MultiProtocol Label Switching Header, Label: 123105, Exp: 0, S: 1, TTL: 255
    0001 1110 0000 1110 0001 .... .... .... = MPLS Label: 123105
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1111 = MPLS TTL: 255
▸ Internet Protocol Version 4, Src: 192.168.3.31, Dst: 192.168.6.61
▸ Internet Control Message Protocol
```

```
CER7#trace 192.168.3.31 sou 192.168.7.71
Type escape sequence to abort.
Tracing the route to 192.168.3.31
VRF info: (vrf in name/id, vrf out name/id)
  1 20.40.27.2 5 msec 2 msec 2 msec
  2 12.39.23.3 10 msec *  10 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 9: MPLS Inter-AS L3VPN

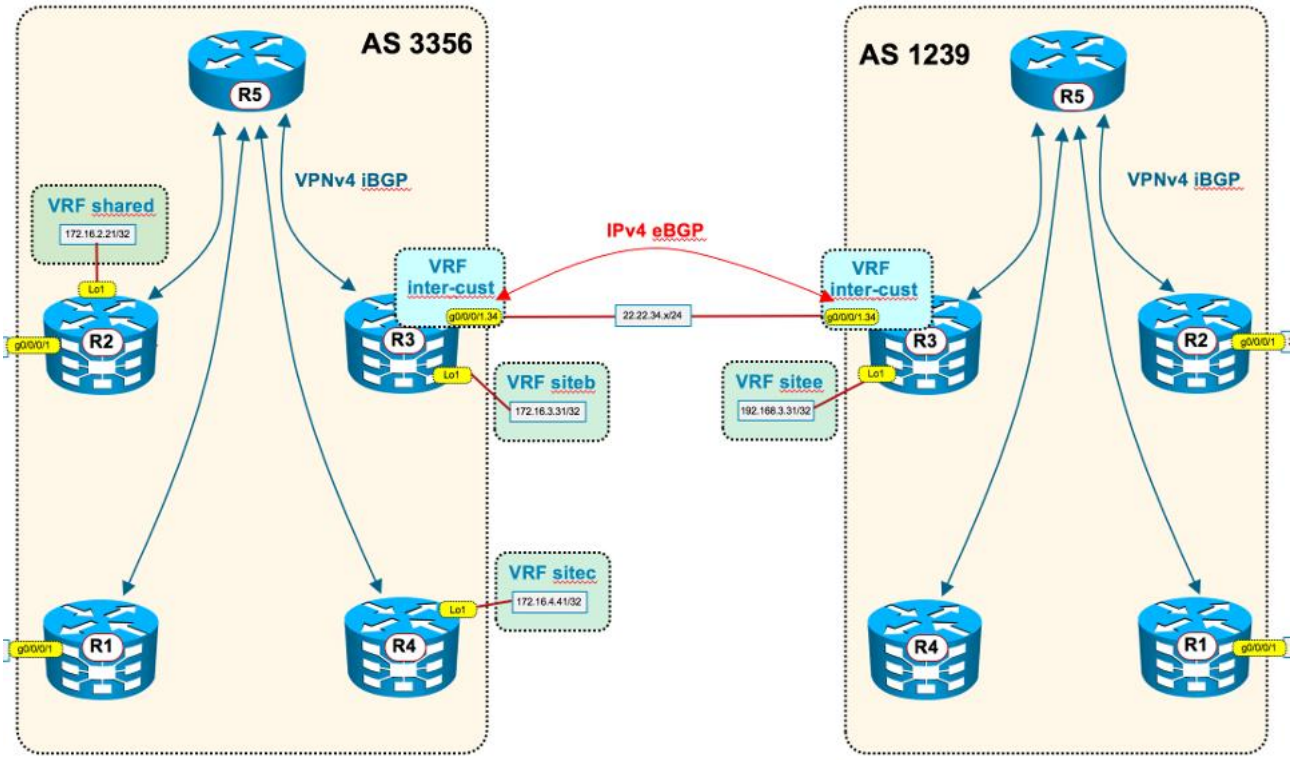
We're going to connect all sites together between AS 3356 and AS 1239.

This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch09.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch09.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch09.unl`

Task 1: Inter-AS option A

Use the Inter-AS option A to exchange L3VPN services between AS domains.



Task 1.1: Build inter-AS connectivity

In the Inter-AS option A the connectivity is built based on a separate IPv4 eBGP session for each of the respective tenant between ASBRs. In our case this tenant is Customer1 represented by several VRFs and sites. All VRFs and sites for Customer1 will be aggregated into one VRF named inter-cust1 on the inter-AS link between AS 3356 and AS 1239.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

A separation between tenants can be done using a separate subinterface or a physical interface with different VRF assigned to them. You will use the separate subinterface between R3 in AS 3356 and R3 in AS 1239.

Configuration of R3 in AS 3356.

```
RP/0/0/CPU0:SP1R3(config)#vrf inter-cust1
RP/0/0/CPU0:SP1R3(config-vrf)#add ipv4 uni
RP/0/0/CPU0:SP1R3(config-vrf-af)#import route-target 1:100
RP/0/0/CPU0:SP1R3(config-vrf-af)#export route-target 1:100
RP/0/0/CPU0:SP1R3(config-vrf-af)#int g0/0/0/1.34
RP/0/0/CPU0:SP1R3(config-subif)#vrf inter-cust1
RP/0/0/CPU0:SP1R3(config-subif)#ipv4 add 22.22.34.1/24
RP/0/0/CPU0:SP1R3(config-subif)#router bgp 3356
RP/0/0/CPU0:SP1R3(config-bgp)#vrf inter-cust1
RP/0/0/CPU0:SP1R3(config-bgp-vrf)#rd 1:100
RP/0/0/CPU0:SP1R3(config-bgp-vrf)#add ipv4 uni
RP/0/0/CPU0:SP1R3(config-bgp-vrf-af)#nei 22.22.34.254
RP/0/0/CPU0:SP1R3(config-bgp-vrf-nbr)#remote-as 1239
RP/0/0/CPU0:SP1R3(config-bgp-vrf-nbr)#add ipv4 uni
RP/0/0/CPU0:SP1R3(config-bgp-vrf-nbr-af)#route-policy PASS in
RP/0/0/CPU0:SP1R3(config-bgp-vrf-nbr-af)#route-policy PASS out
```

Configure R3 in AS 1239. To speed up you can copy and modify the configuration from AS 3356 in a notepad.

```
RP/0/0/CPU0:SP1R3#sh configuration commit change last 1
```

Paste the modified configuration. Differences between configurations are highlighted as follows.

```
RP/0/0/CPU0:SP2R3(config)#vrf inter-cust1
RP/0/0/CPU0:SP2R3(config-vrf)# address-family ipv4 unicast
RP/0/0/CPU0:SP2R3(config-vrf-af)#  import route-target
RP/0/0/CPU0:SP2R3(config-vrf-import-rt)#   2:2
RP/0/0/CPU0:SP2R3(config-vrf-import-rt)#  !
RP/0/0/CPU0:SP2R3(config-vrf-import-rt)#  export route-target
RP/0/0/CPU0:SP2R3(config-vrf-export-rt)#   2:2
RP/0/0/CPU0:SP2R3(config-vrf-export-rt)#  !
RP/0/0/CPU0:SP2R3(config-vrf-export-rt)# !
RP/0/0/CPU0:SP2R3(config-vrf-export-rt)#!
RP/0/0/CPU0:SP2R3(config-vrf-export-rt)#interface GigabitEthernet0/0/0/1.34
RP/0/0/CPU0:SP2R3(config-subif)# vrf inter-cust1
RP/0/0/CPU0:SP2R3(config-subif)# ipv4 address 22.22.34.254 255.255.255.0
RP/0/0/CPU0:SP2R3(config-subif)#!
RP/0/0/CPU0:SP2R3(config-subif)#router bgp 1239
RP/0/0/CPU0:SP2R3(config-bgp)# vrf inter-cust1
RP/0/0/CPU0:SP2R3(config-bgp-vrf)#  rd 200:3356
RP/0/0/CPU0:SP2R3(config-bgp-vrf)#  address-family ipv4 unicast
RP/0/0/CPU0:SP2R3(config-bgp-vrf-af)#  !
RP/0/0/CPU0:SP2R3(config-bgp-vrf-af)#  neighbor 22.22.34.1
RP/0/0/CPU0:SP2R3(config-bgp-vrf-nbr)#   remote-as 3356
RP/0/0/CPU0:SP2R3(config-bgp-vrf-nbr)#   address-family ipv4 unicast
RP/0/0/CPU0:SP2R3(config-bgp-vrf-nbr-af)#    route-policy PASS in
RP/0/0/CPU0:SP2R3(config-bgp-vrf-nbr-af)#    route-policy PASS out
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: Verification

The eBGP session is still down.

```
RP/0/0/CPU0:SP1R3#sh bgp vrf inter-cust1 summ
BGP VRF inter-cust1, state: Active
BGP Route Distinguisher: 1:100
VRF ID: 0x60000007
BGP router identifier 33.56.3.3, local AS number 3356
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000016   RD version: 193
BGP main routing table version 193
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0

BGP is operating in STANDALONE mode.


Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker             193        193        193        193         193           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
22.22.34.254      0  1239       0       0        0    0    0 00:00:00 Idle
```

The interface is also down. What is the reason?

```
RP/0/0/CPU0:SP1R3#sh ip vrf all int brief

Interface                  IP-Address      Status          Protocol Vrf-Name
Loopback0                  33.56.3.3       Up              Up       default
Loopback1                  172.16.3.31     Up              Up       siteb
MgmtEth0/0/CPU0/0          10.255.0.202    Up              Up       Mgmt-intf
GigabitEthernet0/0/0/0     33.56.23.3      Up              Up       default
GigabitEthernet0/0/0/1     22.22.33.1      Up              Up       default
GigabitEthernet0/0/0/1.34  22.22.34.1      Up              Down     inter-cust1
GigabitEthernet0/0/0/2     33.56.34.3      Up              Up       default
GigabitEthernet0/0/0/3     33.56.35.3      Up              Up       default
GigabitEthernet0/0/0/4     10.0.128.1      Up              Up       default
GigabitEthernet0/0/0/5     22.42.37.3      Up              Up       default
GigabitEthernet0/0/0/6     33.56.13.3      Up              Up       default
```

To have subinterfaces up and running you have to have the encapsulation configured.

```
RP/0/0/CPU0:SP1R3(config)#int GigabitEthernet0/0/0/1.34
RP/0/0/CPU0:SP1R3(config-subif)#encapsulation dot1q 34
```

Now you can see prefixes from the other domain on both routers.

```
RP/0/0/CPU0:SP1R3#sh bgp vrf inter-cust1
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:100 (default for vrf inter-cust1)
*>i44.44.67.0/24     33.56.1.1                0    100      0 100 i
* i                  33.56.2.2                0    100      0 100 i
*> 66.66.67.0/24     22.22.34.254                           0 1239 200 i
*>i172.16.2.21/32    33.56.2.2                0    100      0 ?
*>i172.16.6.61/32    33.56.1.1                0    100      0 100 i
* i                  33.56.2.2                     100      0 100 i
*>i172.16.7.71/32    33.56.1.1                     100      0 100 i
* i                  33.56.2.2                0    100      0 100 i
*> 192.168.3.31/32   22.22.34.254                           0 1239 ?
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
*> 192.168.6.61/32    22.22.34.254                           0 1239 200 i
*> 192.168.7.71/32    22.22.34.254                           0 1239 200 i

Processed 8 prefixes, 11 paths


RP/0/0/CPU0:SP2R3#sh bgp vrf inter-cust1
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:3356 (default for vrf inter-cust1)
*> 44.44.67.0/24      22.22.34.1                             0 3356 100 i
* i66.66.67.0/24      12.39.2.2                0    100      0 200 i
*>i                   12.39.1.1                0    100      0 200 i
*> 172.16.2.21/32     22.22.34.1                             0 3356 ?
*> 172.16.6.61/32     22.22.34.1                             0 3356 100 i
*> 172.16.7.71/32     22.22.34.1                             0 3356 100 i
*> 192.168.3.31/32    0.0.0.0                  0         32768 ?
* i192.168.6.61/32    12.39.2.2                     100      0 200 i
*>i                   12.39.1.1                0    100      0 200 i
* i192.168.7.71/32    12.39.2.2                0    100      0 200 i
*>i                   12.39.1.1                     100      0 200 i

Processed 8 prefixes, 11 paths

CEL7#sh ip route bgp
[..]
      66.0.0.0/24 is subnetted, 1 subnets
B        66.66.67.0 [20/0] via 40.60.27.2, 00:04:01
      172.16.0.0/32 is subnetted, 7 subnets
B        172.16.2.21 [20/0] via 40.60.27.2, 20:46:22
B        172.16.3.31 [20/0] via 40.60.27.2, 18:21:29
B        172.16.4.41 [20/0] via 40.60.27.2, 18:21:25
B        172.16.6.61 [200/0] via 172.16.6.6, 3d09h
      192.168.3.0/32 is subnetted, 1 subnets
B        192.168.3.31 [20/0] via 40.60.27.2, 00:04:01
      192.168.6.0/32 is subnetted, 1 subnets
B        192.168.6.61 [20/0] via 40.60.27.2, 00:04:01
      192.168.7.0/32 is subnetted, 1 subnets
B        192.168.7.71 [20/0] via 40.60.27.2, 00:04:01
```

Reachability verification:

```
CEL7#ping 192.168.7.71 sou 172.16.7.71
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.7.71, timeout is 2 seconds:
Packet sent with a source address of 172.16.7.71
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 22/28/37 ms

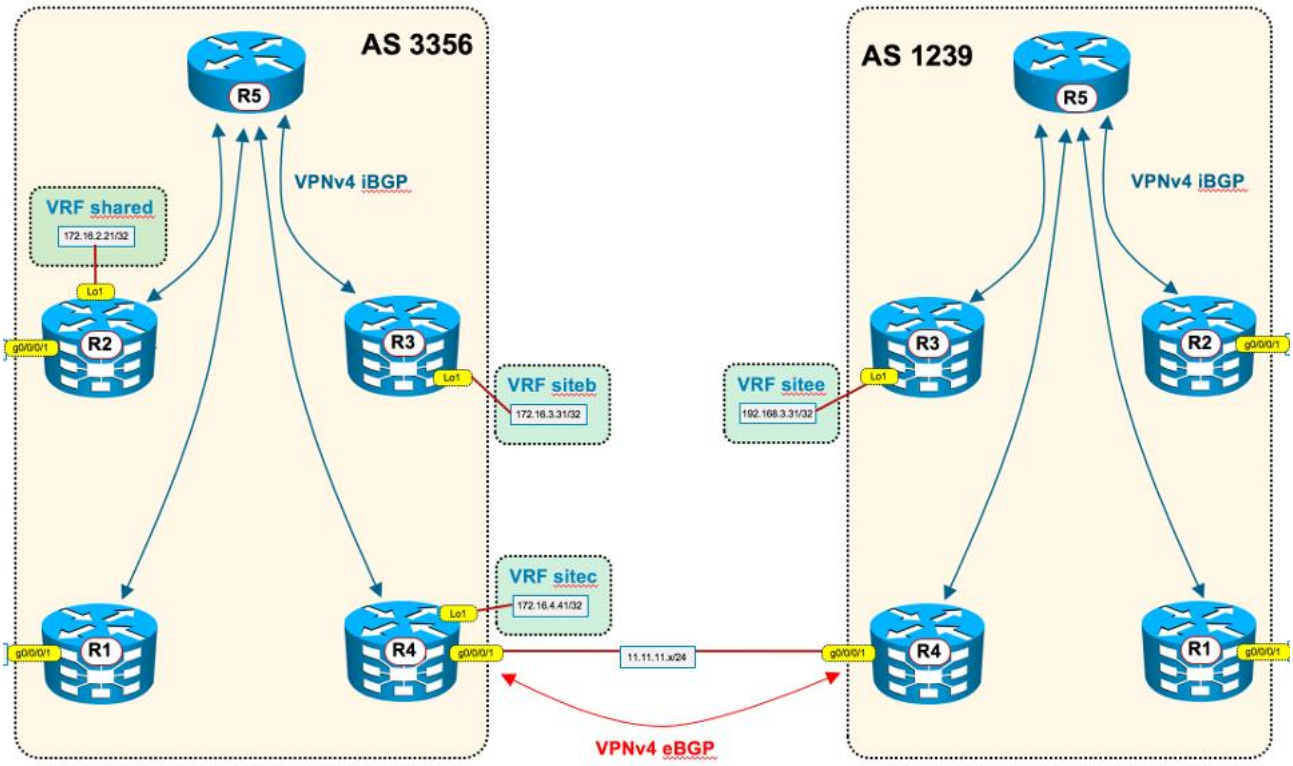
CEL7#trace 192.168.7.71 sou 172.16.7.71
Type escape sequence to abort.
Tracing the route to 192.168.7.71
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 2 msec 2 msec 2 msec
  2 33.56.23.3 [MPLS: Labels 0/103011 Exp 0] 13 msec 11 msec 12 msec
  3 22.22.34.254 11 msec 14 msec 11 msec
  4 12.39.13.1 [MPLS: Labels 0/123106 Exp 0] 37 msec 29 msec 21 msec
  5 21.41.16.6 27 msec 21 msec 21 msec
  6 66.66.67.7 [AS 200] 23 msec *  30 msec
```

There is no visibility to VRFs siteb and sitec from AS 1239. This is done on purpose.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Inter-AS option B

Use Inter-AS option B to exchange L3VPN services between AS 3356 and AS 1239.



Task 2.1: Build inter-AS connectivity

In the Inter-AS option B the connectivity is established based on a VPNv4 eBGP session between ASBRs - SP1R4 and SP2R4 in this case.

Configure ASBRs.

```
RP/0/0/CPU0:SP1R4(config)#router bgp 3356
RP/0/0/CPU0:SP1R4(config-bgp)#nei 11.11.11.254
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#remote-as 1239
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#add vpnv4 uni
RP/0/0/CPU0:SP1R4(config-bgp-nbr-af)#route-policy PASS in
RP/0/0/CPU0:SP1R4(config-bgp-nbr-af)#route-policy PASS out


RP/0/0/CPU0:SP2R4(config)#router bgp 1239
RP/0/0/CPU0:SP2R4(config-bgp)#nei 11.11.11.1
RP/0/0/CPU0:SP2R4(config-bgp-nbr)#remote-as 3356
RP/0/0/CPU0:SP2R4(config-bgp-nbr)#add vpnv4 uni
RP/0/0/CPU0:SP2R4(config-bgp-nbr-af)#route-policy PASS in
RP/0/0/CPU0:SP2R4(config-bgp-nbr-af)#route-policy PASS out
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.2: Prefix advertisement verification

Session are up but there are no prefixes on R4 in AS1239.

```
RP/0/0/CPU0:SP2R4#sh bgp vpnv4 uni summ
[..]

Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker               1          1          1          1           1           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
11.11.11.1        0  3356   16889   15877        1    0    0 00:19:38           0
12.39.5.5         0  1239   19348   16887        1    0    0 02:05:09           0
```

This is caused by the fact that R4 does not import any routes into VRFs. By default, all prefixes all filtered out. You can change this behaviour.

```
RP/0/0/CPU0:SP2R4(config)#router bgp 1239
RP/0/0/CPU0:SP2R4(config-bgp)#add vpnv4 uni
RP/0/0/CPU0:SP2R4(config-bgp-af)#retain ?
  local-label   Delay the release of the local label as configured
  route-target  Accept received updates with the specified route targets
RP/0/0/CPU0:SP2R4(config-bgp-af)#retain route-target ?
  all           Accept received updates containing at least one route target
  route-policy  Accept received updates accepted by the specified policy
RP/0/0/CPU0:SP2R4(config-bgp-af)#retain route-target all
```

Now routes are imported into the VPN table.

```
RP/0/0/CPU0:SP2R4#sh bgp vpnv4 uni
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 3:3
*> 172.16.3.31/32     11.11.11.1                             0 3356 ?
Route Distinguisher: 4:4
*> 172.16.4.41/32     11.11.11.1               0             0 3356 ?
Route Distinguisher: 111:100
*> 44.44.67.0/24      11.11.11.1                             0 3356 100 i
*> 172.16.2.21/32     11.11.11.1                             0 3356 ?
*> 172.16.6.61/32     11.11.11.1                             0 3356 100 i
*> 172.16.7.71/32     11.11.11.1                             0 3356 100 i
Route Distinguisher: 112:100
*> 44.44.67.0/24      11.11.11.1                             0 3356 100 i
*> 172.16.6.61/32     11.11.11.1                             0 3356 100 i
*> 172.16.7.71/32     11.11.11.1                             0 3356 100 i
Route Distinguisher: 200:1
*>i66.66.67.0/24      12.39.2.2                0    100      0 200 i
*>i192.168.6.61/32    12.39.2.2                     100      0 200 i
*>i192.168.7.71/32    12.39.2.2                0    100      0 200 i
Route Distinguisher: 200:2
*>i66.66.67.0/24      12.39.1.1                0    100      0 200 i
*>i192.168.6.61/32    12.39.1.1                0    100      0 200 i
*>i192.168.7.71/32    12.39.1.1                     100      0 200 i
Route Distinguisher: 200:3
*>i192.168.3.31/32    12.39.3.3                0    100      0 ?
Route Distinguisher: 200:3356
*>i44.44.67.0/24      12.39.3.3                     100      0 3356 100 i
*>i172.16.2.21/32     12.39.3.3                     100      0 3356 ?
*>i172.16.6.61/32     12.39.3.3                     100      0 3356 100 i
*>i172.16.7.71/32     12.39.3.3                     100      0 3356 100 i

Processed 20 prefixes, 20 paths
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

But the prefixes learnt by SP2R4 are not received by R2/R1.

```
RP/0/0/CPU0:SP2R2#sh bgp vpnv4 uni rd 200:1
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1 (default for vrf cust1)
*>i44.44.67.0/24      12.39.3.3                   100      0 3356 100 i
* i66.66.67.0/24      12.39.1.1              0    100      0 200 i
*>                    20.40.27.7             0             0 200 i
*>i172.16.2.21/32     12.39.3.3                   100      0 3356 ?
*>i172.16.6.61/32     12.39.3.3                   100      0 3356 100 i
*>i172.16.7.71/32     12.39.3.3                   100      0 3356 100 i
*>i192.168.3.31/32    12.39.3.3              0    100      0 ?
* i192.168.6.61/32    12.39.1.1              0    100      0 200 i
*>                    20.40.27.7                           0 200 i
* i192.168.7.71/32    12.39.1.1                   100      0 200 i
*>                    20.40.27.7             0             0 200 i

Processed 8 prefixes, 11 paths
```

Prefixes are not advertised because on RR those prefixes from R4 are not the best paths as the next hop is not reachable from RR. There is no '>' sign in the BGP table.

```
SP2-R5-RR#sh ip bgp vpnv4 all
[..]
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 3:3
 * i  172.16.3.31/32   11.11.11.1                  100      0 3356 ?
Route Distinguisher: 4:4
 * i  172.16.4.41/32   11.11.11.1              0    100      0 3356 ?
Route Distinguisher: 111:100
 * i  44.44.67.0/24    11.11.11.1                  100      0 3356 100 i
 * i  172.16.2.21/32   11.11.11.1                  100      0 3356 ?
 * i  172.16.6.61/32   11.11.11.1                  100      0 3356 100 i
 * i  172.16.7.71/32   11.11.11.1                  100      0 3356 100 i
Route Distinguisher: 112:100
 * i  44.44.67.0/24    11.11.11.1                  100      0 3356 100 i
 * i  172.16.6.61/32   11.11.11.1                  100      0 3356 100 i
 * i  172.16.7.71/32   11.11.11.1                  100      0 3356 100 i
[..]

SP2-R5-RR#sh ip bgp vpnv4 rd 111:100 172.16.6.61
BGP routing table entry for 111:100:172.16.6.61/32, version 0
Paths: (1 available, no best path)
  Not advertised to any peer
  Refresh Epoch 1
  3356 100, (Received from a RR-client)
    11.11.11.1 (inaccessible) (via default) from 12.39.4.4 (12.39.4.4)
      Origin IGP, localpref 100, valid, internal
      Extended Community: RT:1:100
      mpls labels in/out nolabel/104014
      rx pathid: 0, tx pathid: 0
```

Change the next hop on R4 to itself towards RR in both AS 3356 and AS 1239.

```
RP/0/0/CPU0:SP1R4(config)#router bgp 3356
RP/0/0/CPU0:SP1R4(config-bgp)#nei 33.56.5.5
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#address-family vpnv4 unicast
RP/0/0/CPU0:SP1R4(config-bgp-nbr-af)#next-hop-self
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Now prefixes are valid on RR and also advertised to R2/R1 but some (e.g. 172.16.6.61/32) are still not imported into the VRF cust1.

```
SP2-R5-RR#sh ip bgp vpnv4 all
[..]
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 3:3
 *>i  172.16.3.31/32   12.39.4.4                100      0 3356 ?
Route Distinguisher: 4:4
 *>i  172.16.4.41/32   12.39.4.4                0    100      0 3356 ?
Route Distinguisher: 111:100
 *>i  44.44.67.0/24    12.39.4.4                100      0 3356 100 i
 *>i  172.16.2.21/32   12.39.4.4                100      0 3356 ?
 *>i  172.16.6.61/32   12.39.4.4                100      0 3356 100 i
 *>i  172.16.7.71/32   12.39.4.4                100      0 3356 100 i
Route Distinguisher: 112:100
 *>i  44.44.67.0/24    12.39.4.4                100      0 3356 100 i
 *>i  172.16.6.61/32   12.39.4.4                100      0 3356 100 i
 *>i  172.16.7.71/32   12.39.4.4                100      0 3356 100 i
[..]
```

You can see 172.16.6.61/32 but it is received from SP2R3 not SP2R4.

```
RP/0/0/CPU0:SP2R2#sh bgp vpnv4 uni rd 200:1
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1 (default for vrf cust1)
*>i44.44.67.0/24      12.39.3.3                100      0 3356 100 i
* i66.66.67.0/24      12.39.1.1                0    100      0 200 i
*>                    20.40.27.7               0             0 200 i
* i172.16.2.21/32     12.39.4.4                100      0 3356 ?
*>i                   12.39.3.3                100      0 3356 ?
*>i172.16.6.61/32     12.39.3.3                100      0 3356 100 i
*>i172.16.7.71/32     12.39.3.3                100      0 3356 100 i
*>i192.168.3.31/32    12.39.3.3                0    100      0 ?
* i192.168.6.61/32    12.39.1.1                0    100      0 200 i
*>                    20.40.27.7                        0 200 i
* i192.168.7.71/32    12.39.1.1                100      0 200 i
*>                    20.40.27.7               0             0 200 i

Processed 8 prefixes, 12 paths
```

This is caused by the RT mismatch of original RT export in AS 3356 and RT import settings in VRF in AS 1239.

```
SP2-R5-RR#sh ip bgp vpnv4 rd 111:100 172.16.6.61
BGP routing table entry for 111:100:172.16.6.61/32, version 33
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  3356 100, (Received from a RR-client)
    12.39.4.4 (metric 30) (via default) from 12.39.4.4 (12.39.4.4)
      Origin IGP, localpref 100, valid, internal, best
      Extended Community: RT:1:100
      mpls labels in/out nolabel/123320
      rx pathid: 0, tx pathid: 0x0

RP/0/0/CPU0:SP2R2#sh vrf cust1
VRF                  RD                  RT                          AFI   SAFI
cust1                200:1
                                         import  2:2                 IPV4  Unicast
                                         export  2:2                 IPV4  Unicast
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.3: Traffic engineering

Make the prefixes from R4 more preferable over those learnt from R3 in both AS domains. To do this first of all let's set the matching RT settings. 1:100 in AS 3356 and RT 2:2 in AS 1239 for prefixes learnt for the specific RD. This is the RT rewriting.

Add the configuration on R4 in AS 1239 as follows.

SP2R4:

```
!
route-policy rewrite-cust1
  if rd in (3:3, 4:4, 111:100, 112:100) then
    set extcommunity rt (2:2)
    set local-preference 150
  else
    pass
  endif
end-policy
!

RP/0/0/CPU0:SP2R4(config)#router bgp 1239
RP/0/0/CPU0:SP2R4(config-bgp)#nei 11.11.11.1
RP/0/0/CPU0:SP2R4(config-bgp-nbr)#add vpnv4 uni
RP/0/0/CPU0:SP2R4(config-bgp-nbr-af)#route-policy rewrite-cust1 in
```

Paths over R4 are now preferred.

```
RP/0/0/CPU0:SP2R2#sh bgp vpnv4 uni rd 200:1
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1 (default for vrf cust1)
* i44.44.67.0/24      12.39.4.4                    150      0 3356 100 i
*>i                   12.39.4.4                    150      0 3356 100 i
* i66.66.67.0/24      12.39.1.1                0   100      0 200 i
*>                    20.40.27.7               0            0 200 i
*>i172.16.2.21/32     12.39.4.4                    150      0 3356 ?
*>i172.16.3.31/32     12.39.4.4                    150      0 3356 ?
*>i172.16.4.41/32     12.39.4.4                0   150      0 3356 ?
* i172.16.6.61/32     12.39.4.4                    150      0 3356 100 i
*>i                   12.39.4.4                    150      0 3356 100 i
* i172.16.7.71/32     12.39.4.4                    150      0 3356 100 i
*>i                   12.39.4.4                    150      0 3356 100 i
*>i192.168.3.31/32    12.39.3.3                0   100      0 ?
* i192.168.6.61/32    12.39.1.1                0   100      0 200 i
*>                    20.40.27.7                            0 200 i
* i192.168.7.71/32    12.39.1.1                    100      0 200 i
*>                    20.40.27.7               0            0 200 i

Processed 10 prefixes, 16 paths
```

Do the similar configuration on R4 in AS 3356 to have a preference over R4.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 uni rd 111:100
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 111:100 (default for vrf shared)
* i44.44.67.0/24      33.56.1.1                0   100      0 100 i
*>                    40.60.27.7               0            0 100 i
* i66.66.67.0/24      33.56.4.4                    150      0 1239 200 i
*>i                   33.56.4.4                    150      0 1239 200 i
*> 172.16.2.21/32     0.0.0.0                  0         32768 ?
*>i172.16.3.31/32     33.56.3.3                0   100      0 ?
*>i172.16.4.41/32     33.56.4.4                0   100      0 ?
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
* i172.16.6.61/32     33.56.1.1                0    100      0 100 i
*>                    40.60.27.7                            0 100 i
* i172.16.7.71/32     33.56.1.1                     100      0 100 i
*>                    40.60.27.7               0            0 100 i
*>i192.168.3.31/32    33.56.4.4                     150      0 1239 ?
* i192.168.6.61/32    33.56.4.4                     150      0 1239 200 i
*>i                   33.56.4.4                     150      0 1239 200 i
* i192.168.7.71/32    33.56.4.4                     150      0 1239 200 i
*>i                   33.56.4.4                     150      0 1239 200 i

Processed 10 prefixes, 16 paths
```

The control-plane seems to be fine. Let's verify the data-plane.

```
CEL7#trace 192.168.3.31 source 172.16.7.71
Type escape sequence to abort.
Tracing the route to 192.168.3.31
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 3 msec 2 msec 1 msec
  2  *  *  *
  3
```

Something is wrong with the LSP. R2 shows labels in the FIB table so it should be fine.

```
RP/0/0/CPU0:SP1R2#sh cef vrf cust1 192.168.3.31
192.168.3.31/32, version 297, internal 0x5000001 0x0 (ptr 0xa1927974) [1], 0x0 (0x0),
0x208 (0xa1569230)
 Updated Jan 01 22:57:51.203
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 33.56.4.4/32, 5 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa15d14f4 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 33.56.4.4/32 via 102000/0/21
     next hop 33.56.24.4/32 Gi0/0/0/4    labels imposed {0 104027}
```

Let's check R4 based on the egress label 104027 from the R2's FIB table output. 104027 is the ingress label's value on SP1R4.

```
RP/0/0/CPU0:SP1R4#sh mpls for labels 104027 det
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
104027 123311      200:3:192.168.3.31/32   \
                                                   11.11.11.254    1348
     Updated: Jan 01 22:59:01.900
     Path Flags: 0x6000 [  ]
     Label Stack (Top -> Bottom): { } ! no egress label here
     MAC/Encaps: 0/0, MTU: 0
     Packets Switched: 14

RP/0/0/CPU0:SP1R4#sh mpls ldp bindings 11.11.11.254/32
[empty result]
```

There is no binding to the next hop which is ASBR R4 in AS 1239. This is the expected behaviour by default in IOS XR which does not insert a host /32 route for the ASBR peer. In comparison IOS creates a host route to a peer ASBR.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

You need to create the host route by adding the /32 static route on both R4s.

```
RP/0/0/CPU0:SP1R4(config)#router static
RP/0/0/CPU0:SP1R4(config-static)#add ipv4 uni
RP/0/0/CPU0:SP1R4(config-static-afi)#11.11.11.254/32 g0/0/0/1
```

Add the respective static route on R4 in AS 1239.

Now there is a new label binding.

```
RP/0/0/CPU0:SP1R4#sh mpls ldp bindings 11.11.11.254/32
11.11.11.254/32, rev 18
        Local binding: label: 104020
        No remote bindings
```

There is also an egress label.

```
RP/0/0/CPU0:SP1R4#sh mpls for labels 104027 det
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
104027 123311      200:3:192.168.3.31/32   \
                                     Gi0/0/0/1    11.11.11.254    1348
     Updated: Jan 01 22:59:01.900
     Path Flags: 0x6000 [  ]
     Version: 103, Priority: 4
     Label Stack (Top -> Bottom): { 123311 }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 14/18, MTU: 1500
     Packets Switched: 14
```

Connectivity is established successfully.

```
CEL7#trace 192.168.3.31 source 172.16.7.71
Type escape sequence to abort.
Tracing the route to 192.168.3.31
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 6 msec 1 msec 2 msec
  2 33.56.24.4 [MPLS: Labels 0/104027 Exp 0] 21 msec 27 msec 25 msec
  3 11.11.11.254 [MPLS: Label 123311 Exp 0] 25 msec 19 msec 18 msec
  4 12.39.34.3 19 msec *  30 msec

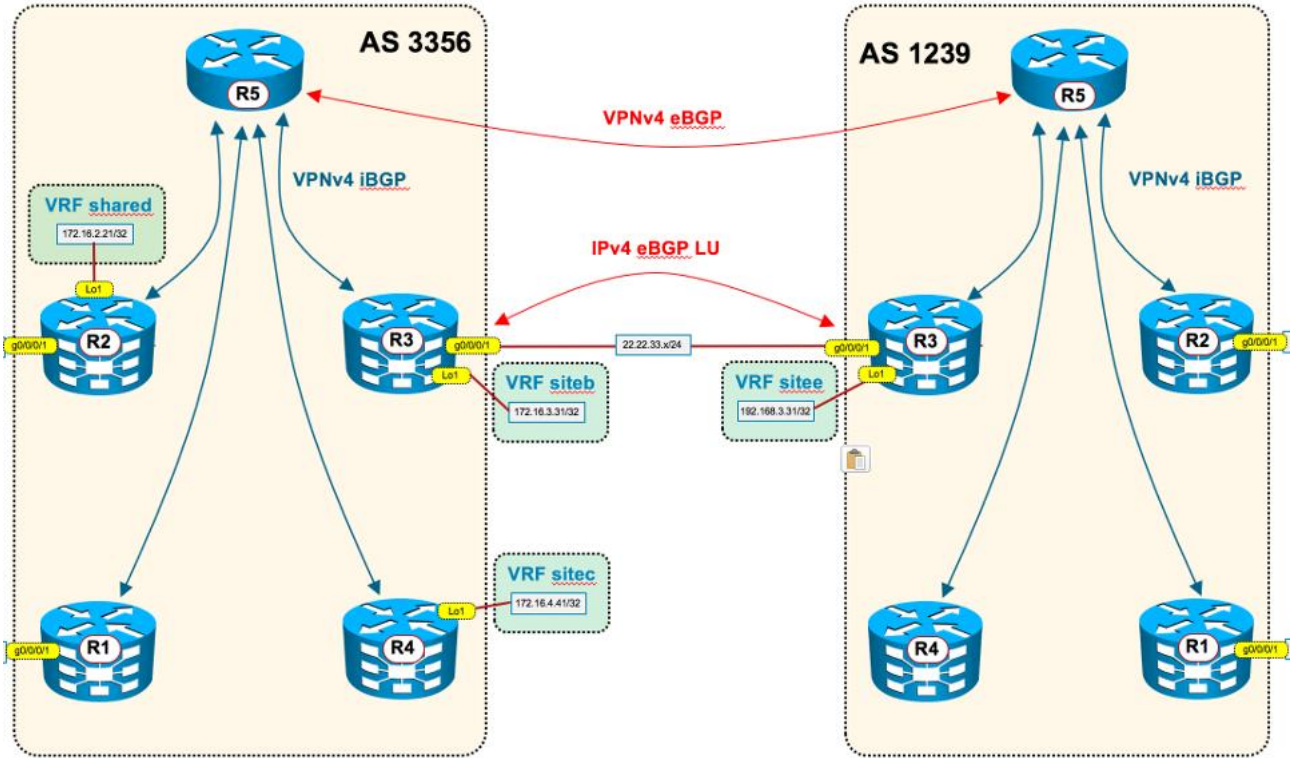
CEL7#trace 192.168.6.61 source 172.16.7.71
Type escape sequence to abort.
Tracing the route to 192.168.6.61
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 3 msec 3 msec 2 msec
  2 33.56.24.4 [MPLS: Labels 0/104025 Exp 0] 28 msec 30 msec 27 msec
  3 11.11.11.254 [MPLS: Label 123309 Exp 0] 46 msec 35 msec 42 msec
  4 12.39.14.1 [MPLS: Labels 0/123105 Exp 0] 32 msec 25 msec 25 msec
  5 21.41.16.6 45 msec *  27 msec

CEL7#trace 192.168.7.71 source 172.16.7.71
Type escape sequence to abort.
Tracing the route to 192.168.7.71
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 2 msec 2 msec 2 msec
  2 33.56.24.4 [MPLS: Labels 0/104026 Exp 0] 54 msec 21 msec 22 msec
  3 11.11.11.254 [MPLS: Label 123310 Exp 0] 23 msec 23 msec 24 msec
  4 12.39.14.1 [MPLS: Labels 0/123106 Exp 0] 20 msec 22 msec 22 msec
  5 21.41.16.6 33 msec 22 msec 22 msec
  6 66.66.67.7 [AS 200] 23 msec *  29 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3: Inter-AS option C

Use Inter-AS option C to exchange L3VPN services between AS 3356 and AS 1239.



Task 3.1: Build Inter-AS connectivity

In the Inter-AS option C the connectivity is established based on a multihop eBGP VPNv4 session between route reflectors SP1R5 and SP2R5.

```
SP1-R5-RR(config)#router bgp 3356
SP1-R5-RR(config-router)#nei 12.39.5.5 remote-as 1239
SP1-R5-RR(config-router)#nei 12.39.5.5 password interas
SP1-R5-RR(config-router)#nei 12.39.5.5 ebgp-multihop
SP1-R5-RR(config-router)#nei 12.39.5.5 update-source lo0
SP1-R5-RR(config-router)#add vpnv4 uni
SP1-R5-RR(config-router-af)#nei 12.39.5.5 activate
SP1-R5-RR(config-router-af)#nei 12.39.5.5 send-community extended
SP1-R5-RR(config-router-af)#nei 12.39.5.5 next-hop-unchanged
```

Leaving a next-hop unchanged is important to have an optimized LSP which is not going through RR. If the next-hop is changed which is a default behaviour then the LSP will be signalled through a route-reflector.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Add a mirrored configuration on RR in AS 1239. Verification.

```
SP1-R5-RR#sh bgp vpnv4 u a su
[..]
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
12.39.5.5       4         1239      11      12      232    0    0 00:01:18        7
33.56.1.1       4         3356    2108    2365      232    0    0 1d11h           3
33.56.2.2       4         3356    2109    2367      232    0    0 1d11h           4
33.56.3.3       4         3356    2119    2362      232    0    0 1d11h           1
33.56.4.4       4         3356    2140    2372      232    0    0 1d11h           8
```

You can see additional paths advertised from AS 1239 with the next hop pointing at the egress PE which is R2 12.39.2.2 in this case.

```
SP1-R5-RR#sh bgp vpnv4 u rd 200:1
[..]

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1
 *    66.66.67.0/24    12.39.2.2                              0 1239 200 i
 *>i                   33.56.4.4                100           0 1239 200 i
 *    192.168.6.61/32  12.39.2.2                              0 1239 200 i
 *>i                   33.56.4.4                100           0 1239 200 i
 *    192.168.7.71/32  12.39.2.2                              0 1239 200 i
 *>i                   33.56.4.4                100           0 1239 200 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2: Traffic-engineering

New paths learnt via the eBGP VPNv4 session from task 3.1 are not advertised further as they are not the best paths. Steer the routing policy in the way that just 172.16.7.71/32 and 192.168.7.71/32 are the routes preferred with the BGP next hop pointing at R2's Loopbacks 33.56.2.2 and 12.39.2.2 respectively.

Before the change.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 u rd 112:100 192.168.7.71/32 brief
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 112:100 (default for vrf cust1)
* i192.168.7.71/32    33.56.4.4                150      0 1239 200 i
*>i                   33.56.4.4                150      0 1239 200 iSP1-R5-RThere
is


RP/0/0/CPU0:SP2R2#sh bgp vpnv4 u rd 200:1 172.16.7.71/32 brief
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1 (default for vrf cust1)
* i172.16.7.71/32     12.39.4.4                150      0 3356 100 i
*>i                   12.39.4.4                150      0 3356 100 i
```

Reachability from AS 3356.

```
CEL7#trace 192.168.7.71 sou 172.16.7.71
Type escape sequence to abort.
Tracing the route to 192.168.7.71
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 2 msec 2 msec 1 msec
  2 33.56.24.4 [MPLS: Labels 0/104026 Exp 0] 26 msec 25 msec 26 msec
  3 11.11.11.254 [MPLS: Label 123310 Exp 0] 23 msec 23 msec 58 msec
  4 12.39.14.1 [MPLS: Labels 0/123106 Exp 0] 33 msec 23 msec 23 msec
  5 21.41.16.6 22 msec 23 msec 27 msec
  6 66.66.67.7 [AS 200] 23 msec *  26 msec
```

Reachability from AS 1239.

```
CER7#trace 172.16.7.71 sou 192.168.7.71
Type escape sequence to abort.
Tracing the route to 172.16.7.71
VRF info: (vrf in name/id, vrf out name/id)
  1 20.40.27.2 3 msec 2 msec 1 msec
  2 12.39.24.4 [MPLS: Labels 0/123324 Exp 0] 34 msec 24 msec 23 msec
  3 11.11.11.1 [MPLS: Label 104018 Exp 0] 21 msec 23 msec 23 msec
  4 33.56.24.2 [MPLS: Labels 0/102010 Exp 0] 24 msec 23 msec 22 msec
  5 40.60.27.7 38 msec *  44 msec
```

Configuration.

```
SP1-R5-RR(config)#ip prefix-list match-r7 permit 192.168.7.71/32
SP1-R5-RR(config)#route-map interas-c
SP1-R5-RR(config-route-map)#match ip add prefix-list match-r7
SP1-R5-RR(config-route-map)#set local-preference 170
SP1-R5-RR(config-route-map)#route-map interas-c 20
SP1-R5-RR(config-route-map)#router bgp 3356
SP1-R5-RR(config-router)#add vpnv4 uni
SP1-R5-RR(config-router-af)#nei 12.39.5.5 route-map interas-c in

SP1-R5-RR#sh route-map interas-c
route-map interas-c, permit, sequence 10
  Match clauses:
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    ip address prefix-lists: match-r7
  Set clauses:
    local-preference 170
  Policy routing matches: 0 packets, 0 bytes
route-map interas-c, permit, sequence 20
  Match clauses:
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
```

Add a respective configuration on RR in AS 1239.

Verification after the config change.

```
SP1-R5-RR#clear ip bgp * soft in
```

Local-preference is changed to 170 as expected.

```
SP1-R5-RR#sh bgp vpnv4 u rd 200:1
[..]

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1
 *    66.66.67.0/24    12.39.2.2                              0 1239 200 i
 *>i                   33.56.4.4                    150       0 1239 200 i
 *    192.168.6.61/32  12.39.2.2                              0 1239 200 i
 *>i                   33.56.4.4                    150       0 1239 200 i
 *>   192.168.7.71/32  12.39.2.2                    170       0 1239 200 i
```

As you can see the BGP next hop is also changed from R4 to R3.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 u rd 112:100 192.168.7.71/32 brief
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 112:100 (default for vrf cust1)
*>i192.168.7.71/32    33.56.3.3                    100       0 1239 200 i

RP/0/0/CPU0:SP2R2#sh bgp vpnv4 u rd 200:1 172.16.7.71/32 brief
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1 (default for vrf cust1)
*>i172.16.7.71/32     12.39.3.3                    100       0 3356 100 i
```

Reachability from AS 3356.

```
CEL7#trace 192.168.7.71 sou 172.16.7.71
Type escape sequence to abort.
Tracing the route to 192.168.7.71
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 3 msec 2 msec 2 msec
  2 33.56.23.3 [MPLS: Labels 0/103007 Exp 0] 11 msec 11 msec 11 msec
  3 22.22.34.254 11 msec 11 msec 10 msec
  4 12.39.13.1 [MPLS: Labels 0/123106 Exp 0] 21 msec 26 msec 22 msec
  5 21.41.16.6 22 msec 21 msec 21 msec
  6 66.66.67.7 [AS 200] 21 msec *  41 msec
```

Reachability from AS 1239.

```
CER7#trace 172.16.7.71 sou 192.168.7.71
Type escape sequence to abort.
Tracing the route to 172.16.7.71
VRF info: (vrf in name/id, vrf out name/id)
  1 20.40.27.2 13 msec 3 msec 2 msec
  2 12.39.23.3 [MPLS: Labels 0/123305 Exp 0] 17 msec 18 msec 14 msec
  3 22.22.34.1 14 msec 15 msec 15 msec
  4 33.56.13.1 21 msec 25 msec 23 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  5 41.61.16.6 27 msec 25 msec 36 msec
  6 44.44.67.7 [AS 100] 29 msec *  36 msec
```

Let's trace the LSP from CEL7 to CER7. There are two labels imposed: transport label 0 (exp-null) and the VPN label 103007. You can see that the BGP next hop is 33.56.3.3 which is resolved to the direct next hop 33.56.23.3.

```
RP/0/0/CPU0:SP1R2#sh cef vrf cust1 192.168.7.71
192.168.7.71/32, version 303, internal 0x5000001 0x0 (ptr 0xa1926f74) [1], 0x0 (0x0),
0x208 (0xa15695c8)
  Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 33.56.3.3/32, 5 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa15d1674 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 33.56.3.3/32 via 102001/0/21
     next hop 33.56.23.3/32 Gi0/0/0/0    labels imposed {0 103007}
```

Then both labels are removed on SP1R3.

```
RP/0/0/CPU0:SP1R3#sh mpls for labels 0
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
0      Aggregate   Exp-Null-v4

RP/0/0/CPU0:SP1R3#sh mpls for
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
103000 Exp-Null-v4 33.56.5.5/32       Gi0/0/0/3    33.56.35.5      28004229
103001 Exp-Null-v4 33.56.2.2/32       Gi0/0/0/0    33.56.23.2      4820595
103002 Exp-Null-v4 33.56.4.4/32       Gi0/0/0/2    33.56.34.4      4826678
103003 Exp-Null-v4 33.56.1.1/32       Gi0/0/0/6    33.56.13.1      2060092
103004 Aggregate   siteb: Per-VRF Aggr[V]   \
                                      siteb                        4716
103005 Aggregate   siteb: Per-VRF Aggr[V]   \
                                      siteb                        1560
103006 Unlabelled  2001:12:39::/48    Gi0/0/0/1    fe80::254       0
103007 Unlabelled  192.168.7.71/32[V] Gi0/0/0/1.34 22.22.34.254    2464
103009 Aggregate   default: Per-VRF Aggr[V]   \
                                      default                      0
```

At this moment the traffic is using the Inter-AS option A path where there are no labels between SP1R3 and SP2R3.

```
RP/0/0/CPU0:SP1R3#sh cef vrf inter-cust1 192.168.7.71
192.168.7.71/32, version 38, internal 0x1000001 0x0 (ptr 0xa18b6cf4) [1], 0x0 (0x0),
0x200 (0xa1569500)
  local adjacency 22.22.34.254
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 22.22.34.254/32, 3 dependencies, recursive, bgp-ext [flags 0x6020]
    path-idx 0 NHID 0x0 [0xa18b70f4 0x0]
    next hop 22.22.34.254/32 via 22.22.34.254/32
     local label 103007
     next hop 22.22.34.254/32 Gi0/0/0/1.34 labels imposed {None}
```

On SP2R3 two labels are pushed again.

```
RP/0/0/CPU0:SP2R3#sh cef vrf inter-cust1 192.168.7.71
192.168.7.71/32, version 4, internal 0x5000001 0x0 (ptr 0xa18a9d74) [1], 0x0 (0x0),
0x208 (0xa1569280)
  Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 12.39.1.1/32, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa15d17f4 0x0]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 12.39.1.1/32 via 123302/0/21
     next hop 12.39.13.1/32 Gi0/0/0/6     labels imposed {0 123106}
```

Labels are removed on R1 in AS 1239 and packets via CER6 go to CER7.

```
RP/0/0/CPU0:SP2R1#sh mpls for
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
123100 Exp-Null-v4 12.39.2.2/32       Gi0/0/0/2    12.39.12.2      4538220
123101 Exp-Null-v4 12.39.5.5/32       Gi0/0/0/3    12.39.15.5      10138629
123102 Exp-Null-v4 12.39.4.4/32       Gi0/0/0/0    12.39.14.4      2066904
123103 Exp-Null-v4 12.39.3.3/32       Gi0/0/0/4    12.39.13.3      2053756
123106 Unlabelled  192.168.7.71/32[V] Gi0/0/0/1    21.41.16.6      11232
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.3: The correct BGP next-hop

In the previous task 3.2 you could notice that even though the preferred BGP path on RR was set to the next hop x.x.2.2 of the egress PE/R2 it was not chosen by the ingress PE. The ASBR as the BGP next hop was chosen instead.

You can see that the path to 192.168.7.71/32 is set via the BGP next hop 12.39.2.2 on RR. The VPN label advertised by SP2R2 is 123206.

```
SP1-R5-RR#sh bgp vpnv4 u rd 200:1 192.168.7.71/32
BGP routing table entry for 200:1:192.168.7.71/32, version 234
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     4
  Refresh Epoch 4
  1239 200
    12.39.2.2 (metric 2) (via default) from 12.39.5.5 (12.39.5.5)
      Origin IGP, localpref 170, valid, external, best
      Extended Community: RT:2:2
      mpls labels in/out 105006/123206
      rx pathid: 0, tx pathid: 0x0
```

This is in line with the label advertised by the egress PE -> SP2R2.

```
RP/0/0/CPU0:SP2R2#sh bgp vpnv4 u labels | i "Network | 123206"
Network            Next Hop        Rcvd Label      Local Label
* i192.168.7.71/32    12.39.1.1       123106          123206
*>                    20.40.27.7      nolabel         123206
```

What is inserted into the VPN table of cust1 on SP1R2?

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 u rd 112:100 192.168.7.71
BGP routing table entry for 192.168.7.71/32, Route Distinguisher: 112:100
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                374         374
Last Modified: Jan 01 12:48:37.950 for 00:59:26
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  1239 200
    33.56.3.3 (metric 2) from 33.56.5.5 (33.56.3.3)
      Received Label 103007
      Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
imported
      Received Path ID 0, Local Path ID 0, version 374
      Extended community: RT:1:100
      Originator: 33.56.3.3, Cluster list: 33.56.5.5
      Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 1:100
```

As you can see the prefix was imported from the VRF inter-cust1 with RD 1:100 received from SP1R3. But the expected result is to have the route imported from the VRF cust1 with RD 200:1 received from SP2R2.

What happened? A modified policy on RR in the task 3.2 removed the prefix advertised by SP1R4 as the prefix from SP2R2 won with the higher local preference 170 vs 150. But the prefix learnt from SP2R2 has RT 2:2 which is not on the import list of SP1R2. So this is a classic situation when two quarrel, a third wins.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

In the Inter-AS L3VPN option C both RD and RT settings must be agreed between different service providers to import prefixes correctly.

On RR there are 3 different VPNv4 prefixes related to 192.168.7.71/32.

```
SP1-R5-RR#sh bgp vpnv4 u a | i Distinguisher|192.168.7.71
Route Distinguisher: 1:100
 *>i  192.168.7.71/32  33.56.3.3                    100      0 1239 200 i
Route Distinguisher: 3:3
Route Distinguisher: 4:4
Route Distinguisher: 111:100
Route Distinguisher: 112:100
Route Distinguisher: 200:1
 *>   192.168.7.71/32  12.39.2.2                    170      0 1239 200 i
Route Distinguisher: 200:2
 *>   192.168.7.71/32  12.39.1.1                    170      0 1239 200 i
Route Distinguisher: 200:3
```

This route is to be imported on SP1R2.

```
SP1-R5-RR#sh bgp vpnv4 u rd 200:1 192.168.7.71/32
BGP routing table entry for 200:1:192.168.7.71/32, version 234
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     4
  Refresh Epoch 4
  1239 200
    12.39.2.2 (metric 2) (via default) from 12.39.5.5 (12.39.5.5)
      Origin IGP, localpref 170, valid, external, best
      Extended Community: RT:2:2
      mpls labels in/out 105006/123206
      rx pathid: 0, tx pathid: 0x0
```

There is no RT 2:2 on the list of SP1R2.

```
RP/0/0/CPU0:SP1R2#sh vrf cust1
VRF                  RD                  RT                          AFI   SAFI
cust1                112:100
                                         import  3:3                  IPV4  Unicast
                                         import  4:4                  IPV4  Unicast
                                         import  1:100                IPV4  Unicast
                                         export  1:100                IPV4  Unicast
                                         import  1:100                IPV6  Unicast
                                         export  1:100                IPV6  Unicast
```

This is not the expected result of the task 3.2. Let's configure it properly to utilize the preferred BGP next hop which is the egress PE not ASBR.

```
RP/0/0/CPU0:SP1R2(config)#vrf cust1
RP/0/0/CPU0:SP1R2(config-vrf)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-vrf-af)#import route-target
RP/0/0/CPU0:SP1R2(config-vrf-import-rt)#2:2

RP/0/0/CPU0:SP2R2(config)#vrf cust1
RP/0/0/CPU0:SP2R2(config-vrf)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R2(config-vrf-af)#import route-target
RP/0/0/CPU0:SP2R2(config-vrf-import-rt)#1:100
```

The BGP next hop is now changed.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 u rd 112:100 192.168.7.71/32 brief

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 112:100 (default for vrf cust1)
* i192.168.7.71/32    33.56.3.3                100      0 1239 200 i
* i                   12.39.2.2                0    170      0 1239 200 i
*>i                   12.39.1.1                0    170      0 1239 200 i
```

This time the BGP next-hop 12.39.1.1 is chosen not 12.39.2.2 on R2 in AS 3356. But in AS 1239 there is a BGP as expected. In your pod it can be different.

SP2R2 has the desired next hop 33.56.2.2.

```
RP/0/0/CPU0:SP2R2#sh bgp vpnv4 u rd 200:1 172.16.7.71/32 brief

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1 (default for vrf cust1)
* i172.16.7.71/32     33.56.1.1                0    170      0 3356 100 i
*>i                   33.56.2.2                0    170      0 3356 100 i
* i                   12.39.3.3                     100      0 3356 100 i
```

On SP1R1 the BGP next hop is 12.39.1.1 which is not the desired state by our task goal. With '**bestpath-compare**' you can check the reason why the path via 12.39.1.1 was taken by SP1R2:

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 u rd 112:100 192.168.7.71/32 bestpath-compare
[…]
Paths: (3 available, best #3)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  1239 200
    33.56.3.3 (metric 2) from 33.56.5.5 (33.56.3.3)
      Received Label 103007
      Origin IGP, localpref 100, valid, internal, imported
      Received Path ID 0, Local Path ID 0, version 0
      Extended community: RT:1:100
      Originator: 33.56.3.3, Cluster list: 33.56.5.5
      Lower local preference than best path (path #3)
      Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 1:100
  Path #2: Received by speaker 0
  Not advertised to any peer
  1239 200
    12.39.2.2 (metric 2) from 33.56.5.5 (33.56.5.5)
      Received Label 123206
      Origin IGP, metric 0, localpref 170, valid, internal, import-candidate, imported
      Received Path ID 0, Local Path ID 0, version 0
      Extended community: RT:2:2
      Higher neighbor address than best path (path #3)
      Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 200:1
  Path #3: Received by speaker 0
  Not advertised to any peer
  1239 200
    12.39.1.1 (metric 2) from 33.56.5.5 (33.56.5.5)
      Received Label 123106
      Origin IGP, metric 0, localpref 170, valid, internal, best, group-best, import-
candidate, imported
      Received Path ID 0, Local Path ID 0, version 399
      Extended community: RT:2:2
      best of AS 1239, Overall best
      Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 200:2
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

12.39.1.1 was chosen over 12.39.2.2 because of the lower neighbor address. Let's modify a routing policy on RRs to choose the path in a deterministic way.

```
RP/0/0/CPU0:SP1R2(config)#vrf cust1 add ipv4 uni export route-target 77:77

RP/0/0/CPU0:SP2R2(config)#vrf cust1 add ipv4 uni export route-target 77:77

SP1-R5-RR(config)#ip extcommunity-list expanded rt-77 permit 77:77
SP1-R5-RR(config)#route-map interas-c 10
SP1-R5-RR(config-route-map)#match extcommunity rt-77

SP2-R5-RR(config)#ip extcommunity-list expanded rt-77 permit 77:77
SP2-R5-RR(config)#route-map interas-c 10
SP2-R5-RR(config-route-map)#match extcommunity rt-77
```

After a configuration modification there is an expected BGP choice.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 u rd 112:100 192.168.7.71/32 brief
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 112:100 (default for vrf cust1)
*>i192.168.7.71/32    12.39.2.2                0    170      0 1239 200 i
* i                   33.56.4.4                     150      0 1239 200 i

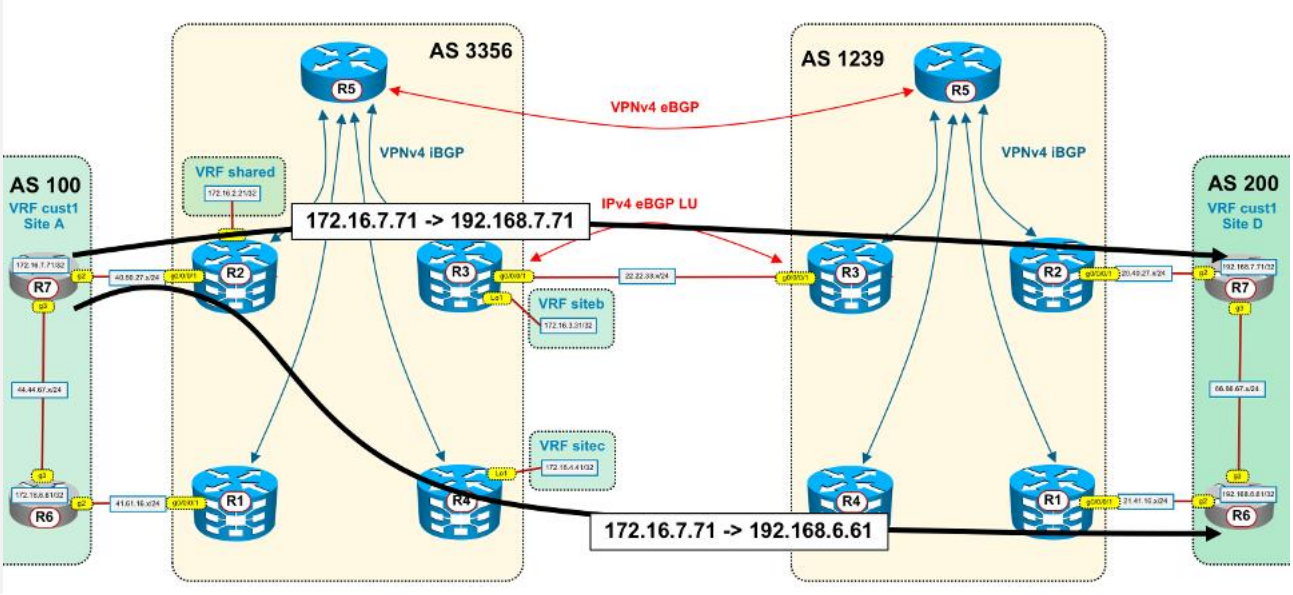
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 u rd 112:100 192.168.7.71/32 bestpath-compare
BGP routing table entry for 192.168.7.71/32, Route Distinguisher: 112:100
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker               423         423
Last Modified: Jan 01 14:36:27.950 for 00:02:44
Paths: (2 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  1239 200
    12.39.2.2 (metric 2) from 33.56.5.5 (33.56.5.5)
      Received Label 123206
      Origin IGP, metric 0, localpref 170, valid, internal, best, group-best, import-
candidate, imported
      Received Path ID 0, Local Path ID 0, version 423
      Extended community: RT:2:2 RT:77:77
      best of AS 1239, Overall best
      Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 200:1
  Path #2: Received by speaker 0
  Not advertised to any peer
  1239 200
    33.56.4.4 (metric 2) from 33.56.5.5 (33.56.4.4)
      Received Label 104026
      Origin IGP, localpref 150, valid, internal, imported
      Received Path ID 0, Local Path ID 0, version 0
      Extended community: RT:1:100
      Originator: 33.56.4.4, Cluster list: 33.56.5.5
      Lower local preference than best path (path #1)
      Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 200:2

RP/0/0/CPU0:SP2R2#sh bgp vpnv4 u rd 200:1 172.16.7.71/32 brief
[..]
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1 (default for vrf cust1)
* i172.16.7.71/32     12.39.4.4                     150      0 3356 100 i
*>i                   33.56.2.2                0    170      0 3356 100 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.4: Continuous LSP

In the task 3.3 you successfully configured the control-plane by advertising the requested prefixes. Now, verify the data-plane. The expected paths are depicted in the following drawing.



Reachability verification.

```
CEL7#trace 192.168.7.71 sou 172.16.7.71
Type escape sequence to abort.
Tracing the route to 192.168.7.71
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 !N  *  !N
```

There is no communication. You can see just the VPN label 123206 advertised by SP2R2 but the BGP next hop 12.39.2.2 is not resolved.

```
RP/0/0/CPU0:SP1R2#sh cef vrf cust1 192.168.7.71
192.168.7.71/32, version 330, internal 0x5000001 0x0 (ptr 0xa1926f74) [1], 0x0 (0x0),
0x208 (0xa15695a0)

 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 12.39.2.2/32, 0 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa0e20294 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    unresolved
     labels imposed {123206}

RP/0/0/CPU0:SP2R2#sh bgp vpnv4 u label | i "Rcvd | 123206"
  Network            Next Hop        Rcvd Label      Local Label
* i192.168.7.71/32    12.39.1.1        123106          123206
*>                    20.40.27.7       nolabel         123206
```

There is no host /32 route to 12.39.2.2 on SP1R2.

```
RP/0/0/CPU0:SP1R2#sh route 12.39.2.2/32

% Network not in table
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Advertise required /32 prefixes. Use the IPv4 eBGP session between AS 3356 and AS 1239 to have a continuous LSP.

SP1R3:

```
route-policy match-r7
  if destination in (33.56.2.2/32) then
    pass
  else
    drop
  endif
end-policy
!
router ospf AS3356
 redistribute bgp 3356 tag 77
!
router bgp 3356
 address-family ipv4 unicast
  no aggregate-address 33.56.0.0/16 summary-only
  allocate-label route-policy match-r7
  redistribute ospf AS3356 route-policy match-r7
 !
 neighbor 22.22.33.254
  no address-family ipv4 unicast
  !
  address-family ipv4 labelled-unicast
   route-policy PASS in
   route-policy PASS out
  !
 !
!
```

SP2R3:

```
!
route-policy match-r7
  if destination in (12.39.2.2/32) then
    pass
  else
    drop
  endif
end-policy
!
router isis AS1239
 address-family ipv4 unicast
  redistribute bgp 1239
 !
!
router bgp 1239
 address-family ipv4 unicast
  no aggregate-address 12.39.0.0/16 summary-only
  allocate-label route-policy match-r7
  redistribute isis AS1239 route-policy match-r7
 !
 neighbor 22.22.33.1
  no address-family ipv4 unicast
  !
  address-family ipv4 labelled-unicast
   route-policy PASS in
   route-policy PASS out
  !
 !
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Verify labels allocated by the eBGP neighbor. These are not the VPN labels but the transport labels. The transport label can be assigned by LDP, RSVP, SR, BGP, static processes.

In the following output 'Network' is the BGP next-hop which we need a label for. 'Next Hop' is an adjacent next-hop on a directly attached link.

```
RP/0/0/CPU0:SP1R3#sh bgp ipv4 labelled-unicast labels
[..]
   Network            Next Hop        Rcvd Label      Local Label
*> 12.39.2.2/32       22.22.33.254    123301          103010
*> 33.56.2.2/32       33.56.23.2      nolabel         103001

Processed 2 prefixes, 2 paths

RP/0/0/CPU0:SP2R3#sh bgp ipv4 labelled-unicast labels
[..]
   Network            Next Hop        Rcvd Label      Local Label
*> 12.39.2.2/32       12.39.23.2      nolabel         123301
*> 33.56.2.2/32       22.22.33.1      103001          123306

Processed 2 prefixes, 2 paths
```

Check if the host route is on R2.

```
RP/0/0/CPU0:SP1R2#sh route 12.39.2.2/32

Routing entry for 12.39.2.2/32
  Known via "ospf AS3356", distance 110, metric 1
  Tag 77, type extern 2
  Installed Jan 01 15:31:04.411 for 00:03:44
  Routing Descriptor Blocks
    33.56.23.3, from 33.56.3.3, via GigabitEthernet0/0/0/0
      Route metric is 1
  No advertising protos.
```

Verify the egress label, the interface and the direct next hop.

```
RP/0/0/CPU0:SP1R2#sh mpls for | i 12.39.2.2
102009 103010      12.39.2.2/32       Gi0/0/0/0    33.56.23.3      0
```

The VRF cust1 FIB table verification.

```
RP/0/0/CPU0:SP1R2#sh cef vrf cust1 192.168.7.71
192.168.7.71/32, version 344, internal 0x5000001 0x0 (ptr 0xa1926f74) [1], 0x0 (0x0),
0x208 (0xa15695a0)

 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 12.39.2.2/32, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa15d1e74 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 12.39.2.2/32 via 102009/0/21
     next hop 33.56.23.3/32 Gi0/0/0/0    labels imposed {103010 123206}
```

The data-plane signalling on SP1R2 seems fine but still it does not work.

```
CEL7#trace 192.168.7.71 sou 172.16.7.71
Type escape sequence to abort.
Tracing the route to 192.168.7.71
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 11 msec 2 msec 2 msec
  2  *  *  *
  3  *  *  *
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Let's check the LSP. From the previous output from SP1R2 you can see that there is the transport label 103010. This label is to be swapped on SP1R3.

```
RP/0/0/CPU0:SP1R3#sh mpls for labels 103010 det
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
103010 123301      12.39.2.2/32                    22.22.33.254    0
     Updated: Jan 01 15:31:59.908
     Path Flags: 0x6020 [  EXT ]
     Label Stack (Top -> Bottom): { }
     MAC/Encaps: 0/0, MTU: 0
     Packets Switched: 0
```

Aha, there is no outgoing interface and no label assigned.

```
RP/0/0/CPU0:SP1R3#sh cef 12.39.2.2
12.39.2.2/32, version 3157, drop adjacency, internal 0x5000001 0x0 (ptr 0xa18b6bf4) [1],
0x0 (0xa14b3a28), 0x808 (0xa1569578)

 Prefix Len 32, traffic index 0, precedence n/a, priority 4
   via 22.22.33.254/32, 0 dependencies, recursive, bgp-ext [flags 0x6020]
    path-idx 0 NHID 0x0 [0xa0e20294 0x0]
    recursion-via-/32
    unresolved
     local label 103010
     labels imposed {123301}
```

The next hop 22.22.33.254 is not resolved which is the same case as in the Inter-AS option B where the host route must be added.

```
RP/0/0/CPU0:SP1R3(config)#router static
RP/0/0/CPU0:SP1R3(config-static)#add ipv4 uni
RP/0/0/CPU0:SP1R3(config-static-afi)#22.22.33.254/32 g0/0/0/1

RP/0/0/CPU0:SP2R3(config)#router static
RP/0/0/CPU0:SP2R3(config-static)#add ipv4 uni
RP/0/0/CPU0:SP2R3(config-static-afi)#22.22.33.1/32 g0/0/0/1
```

Now the label is assigned as the next hop is resolved.

```
RP/0/0/CPU0:SP1R3#sh mpls for labels 103010 det
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
103010 123301      12.39.2.2/32       Gi0/0/0/1    22.22.33.254    0
     Updated: Jan 01 15:55:31.691
     Path Flags: 0x6020 [  EXT ]
     Version: 3157, Priority: 4
     Label Stack (Top -> Bottom): { 123301 }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 14/18, MTU: 1500
     Packets Switched: 0
```

Verify reachability. Destinations should be are reachable.

```
CEL7#trace 192.168.7.71 sou 172.16.7.71
Type escape sequence to abort.
Tracing the route to 192.168.7.71
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 2 msec 2 msec 2 msec
  2 33.56.23.3 [MPLS: Labels 103010/123206 Exp 0] 55 msec 24 msec 22 msec
  3 22.22.33.254 [MPLS: Labels 123301/123206 Exp 0] 21 msec 28 msec 27 msec
  4 12.39.23.2 [MPLS: Labels 0/123206 Exp 0] 24 msec 25 msec 24 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  5 20.40.27.7 27 msec *  27 msec

CER7#trace 172.16.7.71 sou 192.168.7.71
Type escape sequence to abort.
Tracing the route to 172.16.7.71
VRF info: (vrf in name/id, vrf out name/id)
  1 20.40.27.2 2 msec 1 msec 2 msec
  2 12.39.23.3 [MPLS: Labels 123306/102010 Exp 0] 25 msec 50 msec 24 msec
  3 22.22.33.1 [MPLS: Labels 103001/102010 Exp 0] 23 msec 21 msec 21 msec
  4 33.56.23.2 [MPLS: Labels 0/102010 Exp 0] 21 msec 22 msec 24 msec
  5 40.60.27.7 22 msec *  28 msec
```

For other prefixes the path through SP1R4-SP2R4 is maintained.

```
CEL7#trace 192.168.6.61 sou 172.16.7.71
Type escape sequence to abort.
Tracing the route to 192.168.6.61
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 6 msec 2 msec 2 msec
  2 33.56.24.4 [MPLS: Labels 0/104025 Exp 0] 21 msec 20 msec 20 msec
  3 11.11.11.254 [MPLS: Label 123309 Exp 0] 23 msec 23 msec 28 msec
  4 12.39.14.1 [MPLS: Labels 0/123105 Exp 0] 31 msec 21 msec 30 msec
  5 21.41.16.6 33 msec *  23 msec

CER7#trace 172.16.6.61 sou 192.168.7.71
Type escape sequence to abort.
Tracing the route to 172.16.6.61
VRF info: (vrf in name/id, vrf out name/id)
  1 20.40.27.2 3 msec 1 msec 2 msec
  2 12.39.24.4 [MPLS: Labels 0/123323 Exp 0] 30 msec 26 msec 52 msec
  3 11.11.11.1 [MPLS: Label 104017 Exp 0] 23 msec 24 msec 27 msec
  4 33.56.24.2 [MPLS: Labels 0/102011 Exp 0] 25 msec 23 msec 23 msec
  5 40.60.27.7 27 msec 28 msec 28 msec
  6 44.44.67.6 [AS 100] 28 msec *  30 msec
```

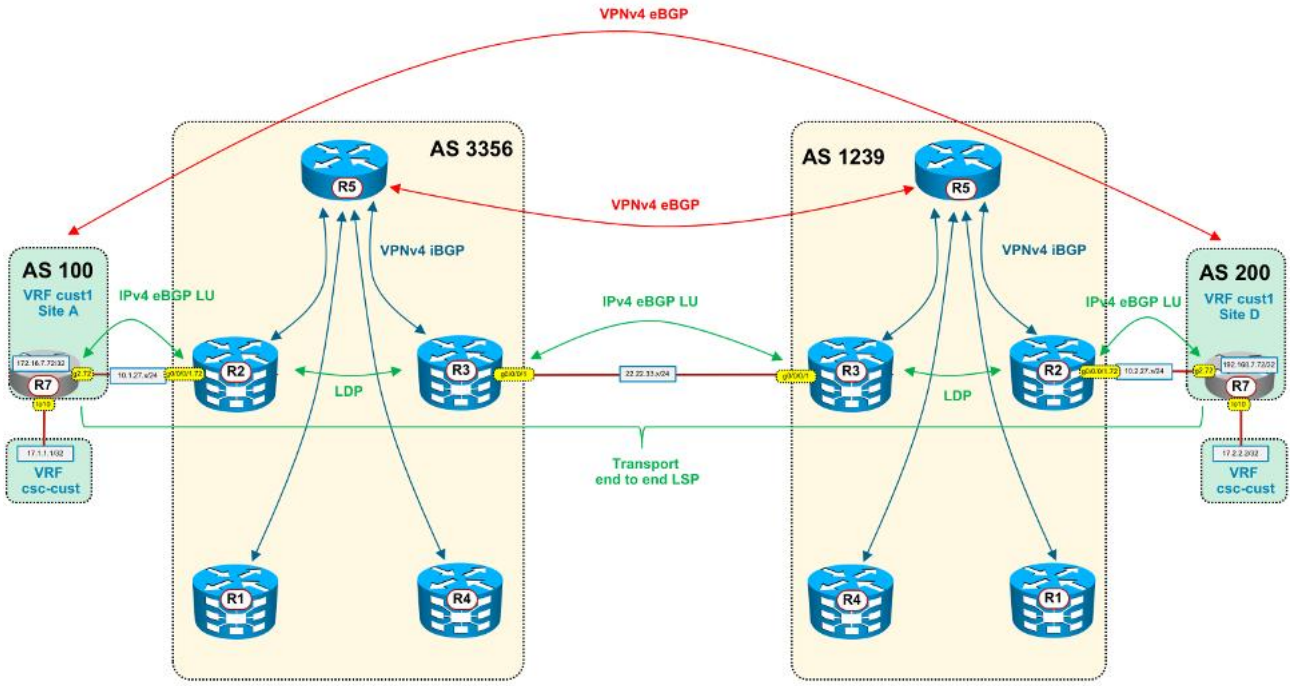
The final result is that the part of the traffic is using the Inter-AS option B and part of it the option C.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4: CSC

The management of Customer1 (AS 100 and AS 200) want to offer L3VPN services for their customers. For the sake of this task AS 100 and AS 200 is acting as a Carrier. Both AS 3356 and AS 1239 serve as a Carrier's Carrier or Carrier Supporting Carrier (CSC).

You will configure the control- and data-plane. Please note that the data plane must be signalled as an end to end labelled switched path between R7 routers (green arrows in the following figure).



© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.1: New subinterfaces and VRF

Configure a new subinterface towards Customer1. Configure the VRF 'csc' in AS 3356 and AS 1239 to have an easy differentiation between CSC and existing L3VPN services. Establish a new IPv4 BGP session + allocate labels between CSC and Customer1.

```
CEL7(config)#int g2.72
CEL7(config-subif)#encapsulation dot1q 72
CEL7(config-subif)#ip add 10.1.27.7 255.255.255.0
CEL7(config-subif)#int lo2
CEL7(config-if)#ip add 172.16.7.72 255.255.255.255
CEL7(config-if)#router bgp 100
CEL7(config-router)#nei 10.1.27.2 remote-as 1239
CEL7(config-router)#add ipv4 uni
CEL7(config-router-af)#nei 10.1.27.2 activate
CEL7(config-router-af)#nei 10.1.27.2 send-label

RP/0/0/CPU0:SP1R2(config)#vrf csc
RP/0/0/CPU0:SP1R2(config-vrf)#add ipv4 uni
RP/0/0/CPU0:SP1R2(config-vrf-af)#import route-target 72:72
RP/0/0/CPU0:SP1R2(config-vrf-af)#export route-target 72:72
RP/0/0/CPU0:SP1R2(config-vrf-af)#int g0/0/0/1.72
RP/0/0/CPU0:SP1R2(config-subif)#encapsulation dot1q 72
RP/0/0/CPU0:SP1R2(config-subif)#vrf csc
RP/0/0/CPU0:SP1R2(config-subif)#ipv4 add 10.1.27.2/24
RP/0/0/CPU0:SP1R2(config-subif)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#vrf csc
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#rd 72:72
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#add ipv4 uni
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#nei 10.1.27.7
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr)#remote-as 100
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr)#add ipv4 labelled-unicast
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr-af)#route-policy PASS in
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr-af)#route-policy PASS out

*Jan 01 20:27:28.599: %BGP-5-ADJCHANGE: neighbor 10.1.27.2 Up
*Jan 01 20:27:28.612: %BGP_LMM-6-AUTOGEN1: The mpls bgp forwarding command has been
configured on interface: GigabitEthernet2.72
```

Note that in IOS XR you don't need to configure 'mpls bgp forwarding' which a required step in IOS and IOS XE.

Configure a routing policy so only one Loopback's IP address is advertised through the new BGP session.

Mark the prefix associated with Loopback2.

```
CEL7(consig)#ip prefix-list PFX-RED-CONN72 permit 172.16.7.72/32
CEL7(config)#route-map CONN->BGP permit 72
CEL7(config-route-map)#match ip address prefix-list PFX-RED-CONN72
CEL7(config-route-map)#set community 72:72
```

Filter out the Lo2's IP address to other neighbors but CSC.

```
CEL7(config)#ip community-list 1 permit 72:72
CEL7(config)#route-map route72
CEL7(config-route-map)#match community 1
CEL7(config)#route-map not-route72 deny 10
CEL7(config-route-map)#match community 1
CEL7(config-route-map)#route-map not-route72 permit 20
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Attach routing policies to BGP neighbors. The CSC neighbor 10.1.27.2 should receive the Lo2's IP address only.

```
CEL7(config-route-map)#router bgp 100
CEL7(config-router)#add ipv4 uni
CEL7(config-router-af)#nei 10.1.27.2 route-map route72 out
CEL7(config-router-af)#nei 40.60.27.2 route-map not-route72 out
CEL7(config-router-af)#nei 172.16.6.6 route-map not-route72 out
```

Due to filtering there is just one prefix in the VPN 72:72 (VRF:csc).

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 uni rd 72:72 labels
[..]
   Network            Next Hop        Rcvd Label      Local Label
Route Distinguisher: 72:72 (default for vrf csc)
*> 172.16.7.72/32     10.1.27.7       3               102019

Processed 1 prefixes, 1 paths
```

Similarly configure the network in AS 200.

```
CER7(config)#int g2.72
CER7(config-subif)#encapsulation dot1q 72
CER7(config-subif)#ip add 10.2.27.7 255.255.255.0
CER7(config-subif)#int lo2
CER7(config-if)#ip add 192.168.7.72 255.255.255.255
CER7(config-if)#router bgp 200
CER7(config-router)#nei 10.2.27.2 remote-as 3356
CER7(config-router)#add ipv4 uni
CER7(config-router-af)#nei 10.2.27.2 activate
CER7(config-router-af)#nei 10.2.27.2 send-label
RP/0/0/CPU0:SP2R2(config)#vrf csc
RP/0/0/CPU0:SP2R2(config-vrf)#add ipv4 uni
RP/0/0/CPU0:SP2R2(config-vrf-af)#import route-target 72:72
RP/0/0/CPU0:SP2R2(config-vrf-af)#export route-target 72:72
RP/0/0/CPU0:SP2R2(config-vrf-af)#int g0/0/0/1.72
RP/0/0/CPU0:SP2R2(config-subif)#encapsulation dot1q 72
RP/0/0/CPU0:SP2R2(config-subif)#vrf csc
RP/0/0/CPU0:SP2R2(config-subif)#ipv4 add 10.2.27.2/24
RP/0/0/CPU0:SP2R2(config-subif)#router bgp 1239
RP/0/0/CPU0:SP2R2(config-bgp)#vrf csc
RP/0/0/CPU0:SP2R2(config-bgp-vrf)#rd 72:72
RP/0/0/CPU0:SP2R2(config-bgp-vrf)#add ipv4 uni
RP/0/0/CPU0:SP2R2(config-bgp-vrf-af)#nei 10.2.27.7
RP/0/0/CPU0:SP2R2(config-bgp-vrf-nbr)#remote-as 200
RP/0/0/CPU0:SP2R2(config-bgp-vrf-nbr)#add ipv4 labelled-unicast
RP/0/0/CPU0:SP2R2(config-bgp-vrf-nbr-af)#route-policy PASS in
RP/0/0/CPU0:SP2R2(config-bgp-vrf-nbr-af)#route-policy PASS out

CER7(config)#ip prefix-list PFX-RED-CONN72 permit 192.168.7.72/32
CER7(config)#route-map CONN->BGP permit 72
CER7(config-route-map)#match ip address prefix-list PFX-RED-CONN72
CER7(config-route-map)#set community 72:72

CER7(config-route-map)#ip community-list 1 permit 72:72
CER7(config)#route-map route72
CER7(config-route-map)#match community 1
CER7(config-route-map)#route-map not-route72 deny 10
CER7(config-route-map)#match community 1
CER7(config-route-map)#route-map not-route72 permit 20

CER7(config)#router bgp 200
CER7(config-router)#add ipv4 uni
CER7(config-router-af)#nei 10.2.27.2 route-map route72 out
CER7(config-router-af)#nei 20.40.27.2 route-map not-route72 out
CER7(config-router-af)#nei 192.168.6.6 route-map not-route72 out
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

After a configuration there are two host /32 routes inside the dedicated VPN.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 uni rd 72:72 labels
[..]
   Network            Next Hop        Rcvd Label      Local Label
Route Distinguisher: 72:72 (default for vrf csc)
*> 172.16.7.72/32     10.1.27.7       3               102019
*>i192.168.7.72/32    12.39.2.2       123212          nolabel

Processed 2 prefixes, 2 paths
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.2: End to end LSP

The task is to build an end to end LSP from CEL7 to CER7. To have this done you need prefixes 172.16.7.72/32 and 192.168.7.72/32 with labels advertised to CER7 and CEL7 respectively.

At this moment there are no prefixes advertised by R2 to R7.

```
CEL7#sh bgp summ
[..]
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.1.27.2       4         3356      73     102     3176    0    0 01:10:36        0
```

This is because a label must be allocated for the IPv4 labelled-unicast address family which is configured on R2.

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#vrf csc
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-vrf-af)#allocate-label all

RP/0/0/CPU0:SP2R2(config)#router bgp 1239
RP/0/0/CPU0:SP2R2(config-bgp)#vrf csc
RP/0/0/CPU0:SP2R2(config-bgp-vrf)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R2(config-bgp-vrf-af)#allocate-label all
```

Verify the settings. The outgoing label on CEL7 and the incoming/local label on SP1R2 must be the same.

```
CEL7#sh bgp 192.168.7.72
BGP routing table entry for 192.168.7.72/32, version 3177
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     21          22
  Refresh Epoch 1
  3356 1239 200
    10.1.27.2 from 10.1.27.2 (33.56.2.2)
      Origin incomplete, localpref 100, valid, external, best
      mpls labels in/out nolabel/102012
      rx pathid: 0, tx pathid: 0x0
```

If R2 routers do not receive x.x.7.72/32 prefixes through the 10.x.27.2 BGP session possibly a filtering does not work in the current XE release. You can change a route-map route72 to a prefix-list. See the config in Chapter 10, Task 0.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 u rd 72:72 labels
[..]
   Network            Next Hop        Rcvd Label      Local Label
Route Distinguisher: 72:72 (default for vrf csc)
*> 172.16.7.72/32     10.1.27.7       3               102019
*>i192.168.7.72/32    12.39.2.2       123212          102012

Processed 2 prefixes, 2 paths
```

The label 102012 is the same on both routers.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Similarly, the outgoing label on CER7 and the incoming/local label on SP2R2 must be the same.

```
CER7#sh bgp 172.16.7.72
BGP routing table entry for 172.16.7.72/32, version 4889
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     6           7
  Refresh Epoch 1
  1239 3356 100
    10.2.27.2 from 10.2.27.2 (12.39.2.2)
      Origin incomplete, localpref 100, valid, external, best
      mpls labels in/out nolabel/123214
      rx pathid: 0, tx pathid: 0x0

RP/0/0/CPU0:SP2R2#sh bgp vpnv4 u rd 72:72 labels
[..]
   Network            Next Hop        Rcvd Label      Local Label
Route Distinguisher: 72:72 (default for vrf csc)
*>i172.16.7.72/32     33.56.2.2       102019          123214
*> 192.168.7.72/32    10.2.27.7       3               123212

Processed 2 prefixes, 2 paths
```

Those labels assigned by BGP are not the VPN but used as transport labels. As in the previous tasks here you need also to add the static routes for host routes of the BGP next hops.

```
RP/0/0/CPU0:SP1R2(config)#router static
RP/0/0/CPU0:SP1R2(config-static)#vrf csc
RP/0/0/CPU0:SP1R2(config-static-vrf)#add ipv4 uni
RP/0/0/CPU0:SP1R2(config-static-vrf-afi)#10.1.27.7/32 g0/0/0/1.72

RP/0/0/CPU0:SP2R2(config)#router static
RP/0/0/CPU0:SP2R2(config-static)#vrf csc
RP/0/0/CPU0:SP2R2(config-static-vrf)#add ipv4 uni
RP/0/0/CPU0:SP2R2(config-static-vrf-afi)#10.2.27.7/32 g0/0/0/1.72
```

Verify labels towards CE which is CEL7 from the SP1R2's perspective. You can see two ImplNull labels. One label is the transport label signalled by CEL7 via eBGP IPv4 session. The other label is the instruction to pop the VPN label 102019 at the edge of the carrier's carrier domain.

```
RP/0/0/CPU0:SP1R2#sh cef vrf csc 172.16.7.72
172.16.7.72/32, version 24, internal 0x1000001 0x0 (ptr 0xa1927774) [1], 0x0 (0x0),
0x200 (0xa1569140)
 Updated Jan 01 20:18:00.682
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 10.1.27.7/32, 3 dependencies, recursive, bgp-ext [flags 0x6020]
    path-idx 0 NHID 0x0 [0xa19266f4 0xa15d1a74]
    recursion-via-/32
    next hop 10.1.27.7/32 via 10.1.27.7/32
     local label 102019
     next hop 10.1.27.7/32 Gi0/0/0/1.72 labels imposed {ImplNull ImplNull}

RP/0/0/CPU0:SP1R2#sh mpls for vrf csc
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
102012 123212      192.168.7.72/32[V]              12.39.2.2       7831
102015 Pop         10.1.27.7/32[V]   Gi0/0/0/1.72 10.1.27.7        12023
102019 Pop         172.16.7.72/32[V] Gi0/0/0/1.72 10.1.27.7        900500
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Test connectivity which verifies the end to end LSP.

```
CEL7#trace 192.168.7.72 sou 172.16.7.72
Type escape sequence to abort.
Tracing the route to 192.168.7.72
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.27.2 [MPLS: Label 102012 Exp 0] 23 msec 25 msec 22 msec
  2 33.56.23.3 [MPLS: Labels 103010/123212 Exp 0] 22 msec 24 msec 21 msec
  3 22.22.33.254 [MPLS: Labels 123301/123212 Exp 0] 24 msec 22 msec 23 msec
  4 12.39.23.2 [MPLS: Labels 0/123212 Exp 0] 20 msec 20 msec 36 msec
  5 10.2.27.7 26 msec *  29 msec

CER7#trace 172.16.7.72 sou 192.168.7.72
Type escape sequence to abort.
Tracing the route to 172.16.7.72
VRF info: (vrf in name/id, vrf out name/id)
  1 10.2.27.2 [MPLS: Label 123214 Exp 0] 27 msec 22 msec 34 msec
  2 12.39.23.3 [MPLS: Labels 123306/102019 Exp 0] 22 msec 23 msec 40 msec
  3 22.22.33.1 [MPLS: Labels 103001/102019 Exp 0] 24 msec 23 msec 32 msec
  4 33.56.23.2 [MPLS: Labels 0/102019 Exp 0] 27 msec 24 msec 22 msec
  5 10.1.27.7 23 msec *  41 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.3: Enable the L3VPN service in AS 100/200

Configure L3VPN service in AS 100 and AS 200.

Define a new VRF and the interface 'csc-cust' which is the customer of AS 100 and AS 200. You can use any RD, RT settings according to your policy. It can be overlapping with settings of the CSC providers.

```
CEL7(config)#vrf definition csc-cust
CEL7(config-vrf)#rd 1:1
CEL7(config-vrf)#add ipv4 uni
CEL7(config-vrf-af)#route-target both 1:1
CEL7(config-vrf-af)#int lo10
CEL7(config-if)#vrf for csc-cust
CEL7(config-if)#ip add 17.1.1.1 255.255.255.255
```

Configure the eBGP VPNv4 session between AS 100 and AS 200. Advertise routes from VRF csc-cust.

```
CEL7(config-if)#router bgp 100
CEL7(config-router)#nei 192.168.7.72 remote-as 200
CEL7(config-router)#nei 192.168.7.72 update-source lo2
CEL7(config-router)#nei 192.168.7.72 ebgp-multihop
CEL7(config-router)#add vpnv4 uni
CEL7(config-router-af)#nei 192.168.7.72 activate
CEL7(config-router-af)#nei 192.168.7.72 send-community extended
CEL7(config-router-af)#add ipv4 uni vrf csc-cust
CEL7(config-router-af)#red conn
```

An equivalent configuration on CER7.

```
CER7(config)#vrf definition csc-cust
CER7(config-vrf)#rd 1:1
CER7(config-vrf)#add ipv4 uni
CER7(config-vrf-af)#route-target both 1:1
CER7(config-vrf-af)#int lo10
CER7(config-if)#vrf for csc-cust
CER7(config-if)#ip add 17.2.2.2 255.255.255.255

CER7(config-if)#router bgp 200
CER7(config-router)#nei 172.16.7.72 remote-as 100
CER7(config-router)#nei 172.16.7.72 update-source lo2
CER7(config-router)#nei 172.16.7.72 ebgp-multihop
CER7(config-router)#add vpnv4 uni
CER7(config-router-af)#nei 172.16.7.72 activate
CER7(config-router-af)#nei 172.16.7.72 send-community extended
CER7(config-router-af)#add ipv4 uni vrf csc-cust
CER7(config-router-af)#red conn
```

Prefix advertisement verification.

```
CEL7#sh bgp vpnv4 un all
BGP table version is 4, local router ID is 172.16.7.7
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf csc-cust)
 *>   17.1.1.1/32      0.0.0.0                  0         32768 ?
 *>   17.2.2.2/32      192.168.7.72             0             0 200 ?
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The VPN label for csc-cust routes.

```
CEL7#sh bgp vpnv4 uni all label
   Network          Next Hop      In label/Out label
Route Distinguisher: 1:1 (csc-cust)
   17.1.1.1/32      0.0.0.0         26/nolabel(csc-cust)
   17.2.2.2/32      192.168.7.72    nolabel/25
```

A verification of FIB table for VRF csc-cust. The VPN label is 25. The transport label is 102012.

```
CEL7#sh ip cef vrf csc-cust 17.2.2.2 det
17.2.2.2/32, epoch 0, flags [rib defined all labels]
  recursive via 192.168.7.72 label 25
    recursive via 10.1.27.2 label 102012
      attached to GigabitEthernet2.72
```

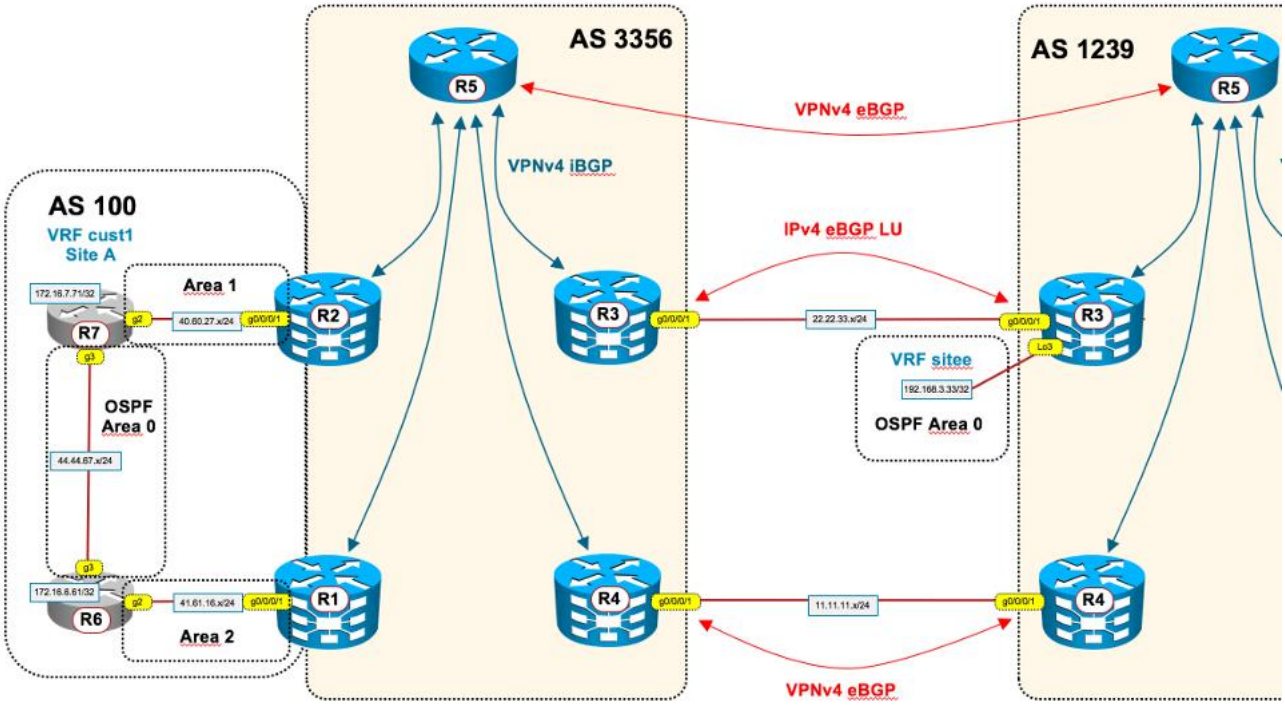
Test your connectivity. You can see that at the border of AS 100 the transport label 102012 is swapped to 103010 and the additional VPN label 123212 is pushed.

```
CEL7#trace vrf csc-cust 17.2.2.2 sou 17.1.1.1
Type escape sequence to abort.
Tracing the route to 17.2.2.2
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.27.2 [MPLS: Labels 102012/25 Exp 0] 24 msec 25 msec 38 msec
  2 33.56.23.3 [MPLS: Labels 103010/123212/25 Exp 0] 42 msec 124 msec 31 msec
  3 22.22.33.254 [MPLS: Labels 123301/123212/25 Exp 0] 26 msec 31 msec 26 msec
  4 12.39.23.2 [MPLS: Labels 0/123212/25 Exp 0] 29 msec 28 msec 34 msec
  5 17.2.2.2 27 msec *  42 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5: OSPF as PE-CE protocol

There is a decision to deploy OSPF at site A (VRF cust1) in AS 100 and site E (VRF: sitee) in AS 200. Connect both sites and exchange routes.



Task 5.1: Deploy OSPF

Configure OSPF accordingly.

```
CEL7(config)#int lo3
CEL7(config-if)#ip add 172.16.7.73 255.255.255.255
CEL7(config-if)#ip ospf 100 area 0
CEL7(config-if)#int g2
CEL7(config-if)#ip ospf 100 area 1

CEL6(config)#int g2
CEL6(config-if)#ip ospf 100 a 2

RP/0/0/CPU0:SP1R2(config)#router ospf 1
RP/0/0/CPU0:SP1R2(config-ospf)#vrf cust1
RP/0/0/CPU0:SP1R2(config-ospf-vrf)#redistribute bgp 3356
RP/0/0/CPU0:SP1R2(config-ospf-vrf)#area 1
RP/0/0/CPU0:SP1R2(config-ospf-vrf-ar)#int g0/0/0/1
RP/0/0/CPU0:SP1R2(config-ospf-vrf-ar-if)#network point-to-point
RP/0/0/CPU0:SP1R2(config-ospf-vrf-ar)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#vrf cust1
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#add ipv4 uni
RP/0/0/CPU0:SP1R2(config-bgp-vrf-af)#redistribute ospf 1

RP/0/0/CPU0:SP1R1(config)#router ospf 1
RP/0/0/CPU0:SP1R1(config-ospf)#vrf cust1
RP/0/0/CPU0:SP1R1(config-ospf-vrf)#redistribute bgp 3356
RP/0/0/CPU0:SP1R1(config-ospf-vrf)#area 2
RP/0/0/CPU0:SP1R1(config-ospf-vrf-ar)#int g0/0/0/1
RP/0/0/CPU0:SP1R1(config-ospf-vrf-ar-if)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#vrf cust1
RP/0/0/CPU0:SP1R1(config-bgp-vrf)#add ipv4 uni
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R1(config-bgp-vrf-af)#redistribute ospf 1

RP/0/0/CPU0:SP2R3(config)#int lo3
RP/0/0/CPU0:SP2R3(config-if)#vrf sitee
RP/0/0/CPU0:SP2R3(config-if)#ipv4 add 192.168.3.33/32
RP/0/0/CPU0:SP2R3(config-if)#router ospf 1
RP/0/0/CPU0:SP2R3(config-ospf)#vrf sitee
RP/0/0/CPU0:SP2R3(config-ospf)#redistribute bgp 1239
RP/0/0/CPU0:SP2R3(config-ospf-vrf)#area 0
RP/0/0/CPU0:SP2R3(config-ospf-vrf-ar)#int lo3
RP/0/0/CPU0:SP2R3(config-ospf-vrf)#router bgp 1239
RP/0/0/CPU0:SP2R3(config-bgp)#vrf sitee
RP/0/0/CPU0:SP2R3(config-bgp-vrf)#add ipv4 uni
RP/0/0/CPU0:SP2R3(config-bgp-vrf-af)#redistribute ospf 1
```

The OSPF session on SP1R2 is up.

```
RP/0/0/CPU0:SP1R2#sh ospf vrf cust1 nei

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF 1, VRF cust1

Neighbor ID     Pri   State           Dead Time   Address         Interface
172.16.7.7      1     FULL/  -        00:00:35    40.60.27.7      GigabitEthernet0/0/0/1
    Neighbor is up for 00:23:28

Total neighbor count: 1
```

But there are no routes inserted into the RIB.

```
RP/0/0/CPU0:SP1R2#sh route vrf cust1 ospf

% No matching routes found
```

As you can see R2 is the ABR and ASBR router. CEL7 is ABR.

```
RP/0/0/CPU0:SP1R2#sh ospf vrf cust1 data router

            OSPF Router with ID (33.56.2.2) (Process ID 1, VRF cust1)

                Router Link States (Area 1)

  LS age: 1471
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 33.56.2.2
  Advertising Router: 33.56.2.2
  LS Seq Number: 80000003
  Checksum: 0x9ef8
  Length: 48
  Area Border Router
  AS Boundary Router
   Number of Links: 2

    Link connected to: another Router (point-to-point)
     (Link ID) Neighboring Router ID: 172.16.7.7
     (Link Data) Router Interface address: 40.60.27.2
      Number of TOS metrics: 0
       TOS 0 Metrics: 1

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 40.60.27.0
     (Link Data) Network Mask: 255.255.255.0
      Number of TOS metrics: 0
       TOS 0 Metrics: 1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Routing Bit Set on this LSA
  LS age: 1471
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 172.16.7.7
  Advertising Router: 172.16.7.7
  LS Seq Number: 80000003
  Checksum: 0x6abc
  Length: 48
  Area Border Router
   Number of Links: 2

    Link connected to: another Router (point-to-point)
     (Link ID) Neighboring Router ID: 33.56.2.2
     (Link Data) Router Interface address: 40.60.27.7
      Number of TOS metrics: 0
       TOS 0 Metrics: 1

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 40.60.27.0
     (Link Data) Network Mask: 255.255.255.0
      Number of TOS metrics: 0
       TOS 0 Metrics: 1
```

Prefix is learnt by R2 but ignored as this is LSA type-3 received from a different ABR through a non-backbone area.

You can see this summary route.

```
RP/0/0/CPU0:SP1R2#sh ospf vrf cust1 data summ 172.16.7.73

            OSPF Router with ID (33.56.2.2) (Process ID 1, VRF cust1)

                Summary Net Link States (Area 1)

  LS age: 135
  Options: (No TOS-capability, DC)
  LS Type: Summary Links (Network)
  Link State ID: 172.16.7.73 (Summary Network Number)
  Advertising Router: 172.16.7.7
  LS Seq Number: 80000002
  Checksum: 0xf36f
  Length: 28
  Network Mask: /32
        TOS: 0  Metric: 1
```

To accept routes you need to make a backbone area contiguous by configuring a virtual-link between CE and PE routers.

```
RP/0/0/CPU0:SP1R2(config)#router ospf 1 vrf cust1
RP/0/0/CPU0:SP1R2(config-ospf-vrf)#area 1
RP/0/0/CPU0:SP1R2(config-ospf-vrf-ar)#virtual-link 172.16.7.7

CEL7(config)#router ospf 100
CEL7(config-router)#area 1 virtual-link 33.56.2.2
```

A new OSPF session for a virtual-link is established.

```
CEL7#sh ip ospf nei

Neighbor ID     Pri   State           Dead Time   Address         Interface
33.56.2.2         0   FULL/  -        00:00:17    40.60.27.2      OSPF_VL0
172.16.6.6        0   FULL/  -        00:00:39    44.44.67.6      GigabitEthernet3
33.56.2.2         0   FULL/  -        00:00:39    40.60.27.2      GigabitEthernet2
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Now prefixes are accepted by R2.

```
RP/0/0/CPU0:SP1R2#sh route vrf cust1 ospf
O IA 41.61.16.0/24 [110/302] via 40.60.27.7, 00:00:42, GigabitEthernet0/0/0/1
O    172.16.6.6/32 [110/302] via 40.60.27.7, 00:00:42, GigabitEthernet0/0/0/1
O    172.16.7.7/32 [110/2] via 40.60.27.7, 00:00:42, GigabitEthernet0/0/0/1
O    172.16.7.73/32 [110/2] via 40.60.27.7, 00:00:42, GigabitEthernet0/0/0/1
```

Some prefixes are there but why there is no 192.168.3.33/32? This is because the prefix is learnt via BGP and this is the source protocol so redistribution to OSPF does not change the preference on SP1R2.

```
RP/0/0/CPU0:SP1R2#sh route vrf cust1 192.168.3.33

Routing entry for 192.168.3.33/32
  Known via "bgp 3356", distance 200, metric 0
  Tag 1239, type internal
  Installed Jan 01 18:10:59.475 for 01:07:02
  Routing Descriptor Blocks
    33.56.4.4, from 33.56.5.5
      Nexthop in Vrf: "default", Table: "default", IPv4 Unicast, Table Id: 0xe0000000
      Route metric is 0
  No advertising protos.
```

To see this prefix in the RIB of CEL7 shut the R2-CEL7 eBGP session down for a while.

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#vrf cust1
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#neighbor 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr)#shutdown
```

You should see the inter-area prefix.

```
CEL7#sh ip route | i 192.168.3.33
O IA     192.168.3.33 [110/2] via 40.60.27.2, 00:05:14, GigabitEthernet2
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5.2: LSA Type manipulation

Let's change the LSA type of the prefix 192.168.3.33/32 advertised by SP2R3.

You can make it external by either redistributing those routes on SP2R3 by 'redistribute connected' or by using different domain-id parameters. By default, XR does not set domain-id. IOS and IOS XE set domain-id.

```
RP/0/0/CPU0:SP1R2(config)#router ospf 1
RP/0/0/CPU0:SP1R2(config-ospf)#vrf cust1
RP/0/0/CPU0:SP1R2(config-ospf-vrf)#domain-id type 0005 value 000000640200

RP/0/0/CPU0:SP2R3(config)#router ospf 1
RP/0/0/CPU0:SP2R3(config-ospf)#vrf sitee
RP/0/0/CPU0:SP2R3(config-ospf-vrf)#domain-id type 0005 value 000000650200
```

The RIB table output after the change.

```
CEL7#sh ip route | i 192.168.3.33
O E2     192.168.3.33 [110/1] via 40.60.27.2, 00:00:03, GigabitEthernet2
```

Now use the same domain-id.

```
RP/0/0/CPU0:SP2R3(config)#router ospf 1
RP/0/0/CPU0:SP2R3(config-ospf)#vrf sitee
RP/0/0/CPU0:SP2R3(config-ospf-vrf)#domain-id type 0005 value 000000640200
```

You should see the inter-area prefix back again.

```
CEL7#sh ip route | i 192.168.3.33
O IA     192.168.3.33 [110/2] via 40.60.27.2, 00:00:04, GigabitEthernet2
```

How to get intra-area LSA for this prefix? Use sham-links which are virtual-links but in the L3VPN. To create a sham-link the same OSPF area number must be on both sides and you need to use IP addresses as endpoints not router-ids.

Configure a new Loopback which is not part of other areas and is advertised by BGP. Should not be advertised by OSPF.

```
RP/0/0/CPU0:SP1R2(config)#interface Loopback3
RP/0/0/CPU0:SP1R2(config-if)#vrf cust1
RP/0/0/CPU0:SP1R2(config-if)#ipv4 address 192.168.2.21/32
RP/0/0/CPU0:SP1R2(config-if)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#vrf cust1
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-vrf-af)#redistribute connected
```

Configure a sham-link on both ends.

```
RP/0/0/CPU0:SP1R2(config)#router ospf 1
RP/0/0/CPU0:SP1R2(config-ospf)#vrf cust1
RP/0/0/CPU0:SP1R2(config-ospf-vrf)#area 0
RP/0/0/CPU0:SP1R2(config-ospf-vrf-ar)#sham-link 192.168.2.21 192.168.3.31

RP/0/0/CPU0:SP2R3(config)#router ospf 1
RP/0/0/CPU0:SP2R3(config-ospf)#vrf sitee
RP/0/0/CPU0:SP2R3(config-ospf-vrf)#area 0
RP/0/0/CPU0:SP2R3(config-ospf-vrf-ar)#sham-link 192.168.3.31 192.168.2.21
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The sham-link status verification.

```
RP/0/0/CPU0:SP1R2#sh ospf 1 vrf cust1 sham-links

Sham Links for OSPF 1, VRF cust1

Sham Link OSPF_SL0 to address 192.168.3.31 is up
Area 0, source address 192.168.2.21
IfIndex = 2
  Run as demand circuit
  DoNotAge LSA allowed., Cost of using 1
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02:372
    Adjacency State FULL (Hello suppressed)
    Number of DBD retrans during last exchange 0
    Index 2/3, retransmission queue length 0, number of retransmission 1
    First 0(0)/0(0) Next 0(0)/0(0)
    Last retransmission scan length is 2, maximum is 2
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

The session state should be FULL.

```
RP/0/0/CPU0:SP1R2#show ospf 1 vrf cust1 neighbor

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF 1, VRF cust1

Neighbor ID     Pri   State           Dead Time   Address         Interface
12.39.3.3       1     FULL/  -           -        192.168.3.31    OSPF_SL0
    Neighbor is up for 00:02:30
172.16.7.7      1     FULL/  -           -        40.60.27.7      OSPF_VL0
    Neighbor is up for 00:27:44
172.16.7.7      1     FULL/  -        00:00:38    40.60.27.7      GigabitEthernet0/0/0/1
    Neighbor is up for 00:27:44

Total neighbor count: 3
```

As the effect you have the intra-area type of prefix in RIB.

```
CEL7#sh ip route | i 192.168.3.33
O        192.168.3.33 [110/3] via 40.60.27.2, 00:05:50, GigabitEthernet2
```

Verify a reachability.

```
CEL7#trace 192.168.3.33 sou 172.16.7.73
Type escape sequence to abort.
Tracing the route to 192.168.3.33
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 [AS 3356] 2 msec 2 msec 1 msec
  2 33.56.24.4 [MPLS: Labels 0/104011 Exp 0] 20 msec 20 msec 22 msec
  3 11.11.11.254 [MPLS: Label 123326 Exp 0] 21 msec 18 msec 19 msec
  4 12.39.34.3 20 msec *  24 msec
```

The last part in this task is to make the backbone link more preferable for the prefix 172.16.6.6/32. Now the path to 172.16.6.6 (CEL6) is going via a direct link CEL7-CEL6. This can be less preferable in a real life scenario due to a lower bandwidth.

```
CEL7#sh ip route 172.16.6.6
Routing entry for 172.16.6.6/32
  Known via "ospf 100", distance 110, metric 301, type intra area
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Last update from 44.44.67.6 on GigabitEthernet3, 00:15:42 ago
  Routing Descriptor Blocks:
  * 44.44.67.6, from 172.16.6.6, 00:15:42 ago, via GigabitEthernet3
      Route metric is 301, traffic share count is 1
```

Configure a sham-link between R2 and R1.

```
RP/0/0/CPU0:SP1R2(config)#router ospf 1
RP/0/0/CPU0:SP1R2(config-ospf)#vrf cust1
RP/0/0/CPU0:SP1R2(config-ospf-vrf)#area 0
RP/0/0/CPU0:SP1R2(config-ospf-vrf-ar)#sham-link 192.168.2.21 192.168.1.11
```

On R1 you require not only the sham-link but also a virtual-link to connect the super-backbone with area 0 where the prefix 172.16.6.6/32 origins from.

```
RP/0/0/CPU0:SP1R1(config)#int lo3
RP/0/0/CPU0:SP1R1(config-if)#vrf cust1
RP/0/0/CPU0:SP1R1(config-if)#ipv4 add 192.168.1.11/32
RP/0/0/CPU0:SP1R1(config-if)#router ospf 1
RP/0/0/CPU0:SP1R1(config-ospf)#vrf cust1
RP/0/0/CPU0:SP1R1(config-ospf-vrf)#area 0
RP/0/0/CPU0:SP1R1(config-ospf-vrf-ar)#sham-link 192.168.1.11 192.168.2.21
RP/0/0/CPU0:SP1R1(config-ospf-vrf-ar-sl)#area 2
RP/0/0/CPU0:SP1R1(config-ospf-vrf-ar)#virtual-link 172.16.6.6
RP/0/0/CPU0:SP1R1(config-ospf-vrf-ar-vl)#router bgp 3356
RP/0/0/CPU0:SP1R1(config-bgp)#vrf cust1
RP/0/0/CPU0:SP1R1(config-bgp-vrf)#add ipv4 uni
RP/0/0/CPU0:SP1R1(config-bgp-vrf-af)#red con
```

OSPF neighbor verification.

```
RP/0/0/CPU0:SP1R2#show ospf 1 vrf cust1 neighbor

Neighbors for OSPF 1, VRF cust1

Neighbor ID     Pri   State           Dead Time   Address         Interface
12.39.3.3       1     FULL/  -           -        192.168.3.31    OSPF_SL0
    Neighbor is up for 00:22:16
33.56.1.1       1     FULL/  -           -        192.168.1.11    OSPF_SL1
    Neighbor is up for 00:02:09
172.16.7.7      1     FULL/  -           -        40.60.27.7      OSPF_VL0
    Neighbor is up for 00:47:30
172.16.7.7      1     FULL/  -        00:00:33    40.60.27.7      GigabitEthernet0/0/0/1
    Neighbor is up for 00:47:30
Total neighbor count: 4


RP/0/0/CPU0:SP1R1#sh ospf 1 vrf cust1 nei

Neighbors for OSPF 1, VRF cust1

Neighbor ID     Pri   State           Dead Time   Address         Interface
33.56.2.2       1     FULL/  -           -        192.168.2.21    OSPF_SL0
    Neighbor is up for 00:01:50
172.16.6.6      1     FULL/  -        00:00:32    41.61.16.6      OSPF_VL0
    Neighbor is up for 00:00:07
172.16.6.6      1     FULL/DR         00:00:37    41.61.16.6      GigabitEthernet0/0/0/1
    Neighbor is up for 03:05:33

Total neighbor count: 3
```

Now the path should be preferred over the super-backbone network. The prefix 172.16.6.6/32 is advertised from CEL6 as LSA Type-1 through the virtual-link

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

CEL6-SP1R1, then the sham-link SP1R1-SP1R2 and finally through the virtual-link SP1R2-CEL7.

```
CEL7#sh ip route 172.16.6.6
Routing entry for 172.16.6.6/32
  Known via "ospf 100", distance 110, metric 4, type intra area
  Last update from 40.60.27.2 on GigabitEthernet2, 00:00:09 ago
  Routing Descriptor Blocks:
  * 40.60.27.2, from 172.16.6.6, 00:00:09 ago, via GigabitEthernet2
      Route metric is 4, traffic share count is 1

CEL7#trace 172.16.6.6 sou 172.16.7.7
Type escape sequence to abort.
Tracing the route to 172.16.6.6
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 [AS 3356] 3 msec 2 msec 2 msec
  2 33.56.12.1 10 msec 9 msec 8 msec
  3 41.61.16.6 [AS 3356] 10 msec *  15 msec
```

As you can see that through the manipulation of the LSA type in OSPF you can influence the routing decisions.

Please note that in the production environment the IP addresses which are used to span the sham-link should not be redistributed into OSPF. Such a leaking can cause a flapping of sham-links as the prefix can be re-learnt through the OSPF domain instead of the MP-BGP.

Configure a filter for sham-link endpoints on both SP1R2 and SP1R1 to avoid potential issues.

SP1R2, SP1R1:

```
prefix-set sham-filter
  192.168.1.11/32,
  192.168.2.21/32
end-set
!
route-policy sham-filter
  if destination in sham-filter then
    drop
  else
    pass
  endif
end-policy
!
router ospf 1
 vrf cust1
  redistribute bgp 3356 route-policy sham-filter
```

For the sake of this lab remove the sham-link to Site E.

```
RP/0/0/CPU0:SP1R2(config)#router ospf 1
RP/0/0/CPU0:SP1R2(config-ospf)#vrf cust1
RP/0/0/CPU0:SP1R2(config-ospf-vrf)#area 0
RP/0/0/CPU0:SP1R2(config-ospf-vrf-ar)#no sham-link 192.168.2.21 192.168.3.31
```

Bring back the BGP session.

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#vrf cust1
RP/0/0/CPU0:SP1R2(config-bgp-vrf)#neighbor 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-vrf-nbr)#no shutdown
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 6: EIGRP as PE-CE protocol

Deploy EIGRP site D (VRF cust1) in AS 200 and site C (VRF: sitec) in AS 100. Connect both sites and exchange routes.

Task 6.1: Deploy EIGRP

Configure EIGRP as follows.

```
CER7(config)#int lo3
CER7(config-if)#ip add 192.168.7.73 255.255.255.255
CER7(config-if)#router eigrp 100
CER7(config-router)#net 0.0.0.0

CER6(config)#router eigrp 100
CER6(config-router)#net 0.0.0.0

RP/0/0/CPU0:SP2R2(config)#route-policy eigrp-metric
RP/0/0/CPU0:SP2R2(config-rpl)#set eigrp-metric 1000000 1 255 1 1500
RP/0/0/CPU0:SP2R2(config)#router eigrp 1
RP/0/0/CPU0:SP2R2(config-eigrp)#vrf cust1
RP/0/0/CPU0:SP2R2(config-eigrp-vrf)#add ipv4
RP/0/0/CPU0:SP2R2(config-eigrp-vrf-af)#autonomous-system 100
RP/0/0/CPU0:SP2R2(config-eigrp-vrf-af)#redistribute bgp 1239 route-policy eigrp-metric
RP/0/0/CPU0:SP2R2(config-eigrp-vrf-af)#int g0/0/0/1
RP/0/0/CPU0:SP2R2(config-eigrp-vrf-af-if)#router bgp 1239
RP/0/0/CPU0:SP2R2(config-bgp)#vrf cust1
RP/0/0/CPU0:SP2R2(config-bgp-vrf)#add ipv4 uni
RP/0/0/CPU0:SP2R2(config-bgp-vrf-af)#redistribute eigrp 100

RP/0/0/CPU0:SP2R1(config)#router eigrp 1
RP/0/0/CPU0:SP2R1(config-eigrp)#vrf cust1
RP/0/0/CPU0:SP2R1(config-eigrp-vrf)#address-family ipv4
RP/0/0/CPU0:SP2R1(config-eigrp-vrf-af)#autonomous-system 100
RP/0/0/CPU0:SP2R1(config-eigrp-vrf-af)#redistribute bgp 1239
RP/0/0/CPU0:SP2R1(config-eigrp-vrf-af)#interface GigabitEthernet0/0/0/1
RP/0/0/CPU0:SP2R1(config-eigrp-vrf-af-if)#router bgp 1239
RP/0/0/CPU0:SP2R1(config-bgp)#vrf cust1
RP/0/0/CPU0:SP2R1(config-bgp-vrf)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R1(config-bgp-vrf-af)#redistribute eigrp 100 metric 1

RP/0/0/CPU0:SP1R4(config)#int lo3
RP/0/0/CPU0:SP1R4(config-if)#vrf sitec
RP/0/0/CPU0:SP1R4(config-if)#ipv4 add 172.16.4.43/32
RP/0/0/CPU0:SP1R4(config-if)#router eigrp 1
RP/0/0/CPU0:SP1R4(config-eigrp)#vrf sitec
RP/0/0/CPU0:SP1R4(config-eigrp-vrf)#add ipv4
RP/0/0/CPU0:SP1R4(config-eigrp-vrf-af)#autonomous-system 200
RP/0/0/CPU0:SP1R4(config-eigrp-vrf-af)#redistribute bgp 3356
RP/0/0/CPU0:SP1R4(config-eigrp-vrf-af)#int lo3
RP/0/0/CPU0:SP1R4(config-eigrp-vrf-af-if)#router bgp 3356
RP/0/0/CPU0:SP1R4(config-bgp)#vrf sitec
RP/0/0/CPU0:SP1R4(config-bgp-vrf)#add ipv4 uni
RP/0/0/CPU0:SP1R4(config-bgp-vrf-af)#redistribute connected
```

After the configuration the prefix 172.16.4.43/32 from Site C is known via BGP instead of EIGRP in the RIB of CER7. This is because the administrative distance of eBGP (20) is lower than EIGRP (90 or 170).

```
CER7#sh ip route 172.16.4.43
Routing entry for 172.16.4.43/32
  Known via "bgp 200", distance 20, metric 0
  Tag 1239, type external
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Last update from 20.40.27.2 00:05:59 ago
  Routing Descriptor Blocks:
  * 20.40.27.2, from 20.40.27.2, 00:05:59 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2
      Route tag 1239
      MPLS label: none
```

Shut down eBGP session for testing purposes.

```
RP/0/0/CPU0:SP2R1(config)#router bgp 1239
RP/0/0/CPU0:SP2R1(config-bgp)#vrf cust1
RP/0/0/CPU0:SP2R1(config-bgp-vrf)#neighbor 21.41.16.6
RP/0/0/CPU0:SP2R1(config-bgp-vrf-nbr)#shut

RP/0/0/CPU0:SP2R2(config)#router bgp 1239
RP/0/0/CPU0:SP2R2(config-bgp)#vrf cust1
RP/0/0/CPU0:SP2R2(config-bgp-vrf)#neighbor 20.40.27.7
RP/0/0/CPU0:SP2R2(config-bgp-vrf-nbr)#shut
```

Now you can see the prefix known via EIGRP.

```
CER7#sh ip route 172.16.4.43
Routing entry for 172.16.4.43/32
  Known via "eigrp 100", distance 170, metric 3072
  Tag 3356, type external
  Redistributing via eigrp 100
  Last update from 20.40.27.2 on GigabitEthernet2, 00:05:59 ago
  Routing Descriptor Blocks:
  * 20.40.27.2, from 20.40.27.2, 00:05:59 ago, via GigabitEthernet2
      Route metric is 3072, traffic share count is 1
      Total delay is 20 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
      Route tag 3356
```

Try to reach the destination.

```
CER7#trace 172.16.4.43
Type escape sequence to abort.
Tracing the route to 172.16.4.43
VRF info: (vrf in name/id, vrf out name/id)
  1 20.40.27.2 2 msec 4 msec 5 msec
  2 12.39.12.1 [MPLS: Labels 0/123119 Exp 0] 816 msec 839 msec 856 msec
  3 21.41.16.6 38 msec 25 msec 27 msec
  4 66.66.67.7 [MPLS: Label 45 Exp 0] 853 msec 836 msec 794 msec
  5 20.40.27.2 788 msec 807 msec 996 msec
  6 12.39.12.1 [MPLS: Labels 0/123119 Exp 0] 1581 msec 1568 msec 1591 msec
  7 21.41.16.6 803 msec 784 msec 797 msec
  8 66.66.67.7 [MPLS: Label 45 Exp 0] 1496 msec 1479 msec 1448 msec
[..]
```

Oops, you possibly have a loop here.

The VPN table of SP2R2 shows that the BGP next hop is 12.39.1.1 instead of 12.39.4.4.

```
RP/0/0/CPU0:SP2R2#sh bgp vpnv4 uni rd 200:1 172.16.4.43
BGP routing table entry for 172.16.4.43/32, Route Distinguisher: 200:1
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker               572         572
Last Modified: Jan 01 23:04:06.585 for 00:10:16
Paths: (2 available, best #2)
  Not advertised to any peer
  Path #1: Received by speaker 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Not advertised to any peer
  3356
    12.39.4.4 (metric 20) from 12.39.5.5 (12.39.4.4)
      Received Label 123348
      Origin incomplete, metric 0, localpref 150, valid, internal, group-best, imported
      Received Path ID 0, Local Path ID 0, version 0
      Extended community: EIGRP route-info:0x8000:0 EIGRP AD:200:26 EIGRP RHB:255:0:256
EIGRP LM:0x0:1:1514 EIGRP VRR:0x0:4.4.56.33 RT:2:2
      Originator: 12.39.4.4, Cluster list: 12.39.5.5
      Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 4:4
  Path #2: Received by speaker 0
  Not advertised to any peer
  Local
    12.39.1.1 (metric 20) from 12.39.5.5 (12.39.1.1)
      Received Label 123119
      Origin incomplete, metric 1, localpref 100, valid, internal, best, group-best,
import-candidate, imported
      Received Path ID 0, Local Path ID 0, version 572
      Extended community: COST:128:129:25600 EIGRP route-info:0x0:3356 EIGRP AD:100:1024
EIGRP RHB:255:3:2560 EIGRP LM:0x0:1:1500 EIGRP AR:100:12.39.2.2 EIGRP PM:9:0 RT:2:2
      Originator: 12.39.1.1, Cluster list: 12.39.5.5
      Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 200:2
```

Unexpectedly a loop occurred. Asymmetric EIGRP metric settings caused this issue but also a lack of a loop prevention.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 6.2: Loop prevention

Deploy Site of Origin (SoO) as a loop prevention mechanism.

```
RP/0/0/CPU0:SP2R2(config)#router eigrp 1
RP/0/0/CPU0:SP2R2(config-eigrp)#vrf cust1
RP/0/0/CPU0:SP2R2(config-eigrp-vrf)#address-family ipv4
RP/0/0/CPU0:SP2R2(config-eigrp-vrf-af)#interface GigabitEthernet0/0/0/1
RP/0/0/CPU0:SP2R2(config-eigrp-vrf-af-if)#site-of-origin 2:2

RP/0/0/CPU0:SP2R1(config)#router eigrp 1
RP/0/0/CPU0:SP2R1(config-eigrp)#vrf cust1
RP/0/0/CPU0:SP2R1(config-eigrp-vrf)#address-family ipv4
RP/0/0/CPU0:SP2R1(config-eigrp-vrf-af)#interface GigabitEthernet0/0/0/1
RP/0/0/CPU0:SP2R1(config-eigrp-vrf-af-if)#site-of-origin 2:2
```

Verify if the loop is fixed.

```
CER7#trace 172.16.4.43
Type escape sequence to abort.
Tracing the route to 172.16.4.43
VRF info: (vrf in name/id, vrf out name/id)
  1 20.40.27.2 2 msec 1 msec 1 msec
  2 12.39.12.1 [MPLS: Labels 0/123119 Exp 0] 7 msec 15 msec 9 msec
  3 21.41.16.6 7 msec 9 msec 8 msec
  4 66.66.67.7 [MPLS: Label 45 Exp 0] 8 msec 7 msec 8 msec
  5 20.40.27.2 7 msec 7 msec 7 msec
  6 12.39.12.1 [MPLS: Labels 0/123119 Exp 0] 12 msec 11 msec 19 msec
  7 21.41.16.6 12 msec 12 msec 12 msec
  8 66.66.67.7 [MPLS: Label 45 Exp 0] 12 msec 12 msec 12 msec
[..]
```

As it occurs SoO does not help in this case. Maybe the SoO community is not set?

```
RP/0/0/CPU0:SP2R2#sh bgp vpnv4 uni rd 200:1 172.16.4.43
BGP routing table entry for 172.16.4.43/32, Route Distinguisher: 200:1
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                822         822
Last Modified: Jan 01 23:23:34.585 for 00:04:48
Paths: (2 available, best #2)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  3356
    12.39.4.4 (metric 20) from 12.39.5.5 (12.39.4.4)
      Received Label 123348
      Origin incomplete, metric 0, localpref 150, valid, internal, group-best, imported
      Received Path ID 0, Local Path ID 0, version 0
      Extended community: EIGRP route-info:0x8000:0 EIGRP AD:200:26 EIGRP RHB:255:0:256
EIGRP LM:0x0:1:1514 EIGRP VRR:0x0:4.4.56.33 RT:2:2
      Originator: 12.39.4.4, Cluster list: 12.39.5.5
      Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 4:4
  Path #2: Received by speaker 0
  Not advertised to any peer
  Local
    12.39.1.1 (metric 20) from 12.39.5.5 (12.39.1.1)
      Received Label 123119
      Origin incomplete, metric 1, localpref 100, valid, internal, best, group-best,
import-candidate, imported
      Received Path ID 0, Local Path ID 0, version 822
      Extended community: SoO:2:2 COST:128:129:25600 EIGRP route-info:0x0:3356 EIGRP
AD:100:1024 EIGRP RHB:255:3:2560 EIGRP LM:0x0:1:1500 EIGRP AR:100:12.39.2.2 EIGRP PM:9:0
RT:2:2
      Originator: 12.39.1.1, Cluster list: 12.39.5.5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 200:2
```

No, the SoO community is set correctly.

Let's try to use tags to match and filter out the prefix at the edge of L3VPN.

```
CER7#sh ip route 172.16.4.43
Routing entry for 172.16.4.43/32
  Known via "eigrp 100", distance 170, metric 3072
  Tag 3356, type external
  Redistributing via eigrp 100
  Last update from 20.40.27.2 on GigabitEthernet2, 00:12:09 ago
  Routing Descriptor Blocks:
  * 20.40.27.2, from 20.40.27.2, 00:12:09 ago, via GigabitEthernet2
      Route metric is 3072, traffic share count is 1
      Total delay is 20 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
      Route tag 3356

RP/0/0/CPU0:SP2R1(config)#route-policy match-tag ($tag)
RP/0/0/CPU0:SP2R1(config-rpl)#if tag is $tag then drop else pass endif
RP/0/0/CPU0:SP2R1(config-rpl)#exit
RP/0/0/CPU0:SP2R1(config)#router bgp 1239
RP/0/0/CPU0:SP2R1(config-bgp)#vrf cust1
RP/0/0/CPU0:SP2R1(config-bgp-vrf)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R1(config-bgp-vrf-af)#redistribute eigrp 100 metric 1 route-policy match-
tag(3356)
```

After applying the prefix filtering the destination is reachable through 12.39.4.4.

```
CER7#trace 172.16.4.43
Type escape sequence to abort.
Tracing the route to 172.16.4.43
VRF info: (vrf in name/id, vrf out name/id)
  1 20.40.27.2 5 msec 4 msec 1 msec
  2 12.39.24.4 [MPLS: Labels 0/123348 Exp 0] 18 msec 16 msec 16 msec
  3 11.11.11.1 24 msec *  39 msec

RP/0/0/CPU0:SP2R2#sh bgp vpnv4 uni rd 200:1 172.16.4.43
BGP routing table entry for 172.16.4.43/32, Route Distinguisher: 200:1
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker               867         867
Last Modified: Jan 01 23:38:01.585 for 00:01:50
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  3356
    12.39.4.4 (metric 20) from 12.39.5.5 (12.39.4.4)
      Received Label 123348
      Origin incomplete, metric 0, localpref 150, valid, internal, best, group-best,
import-candidate, imported
      Received Path ID 0, Local Path ID 0, version 867
      Extended community: EIGRP route-info:0x8000:0 EIGRP AD:200:26 EIGRP RHB:255:0:256
EIGRP LM:0x0:1:1514 EIGRP VRR:0x0:4.4.56.33 RT:2:2
      Originator: 12.39.4.4, Cluster list: 12.39.5.5
      Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 4:4
```

There is no BGP path to 172.16.4.43 via SP2R1 on SP2R2 which is the expected result.

Bring back the eBGP sessions.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

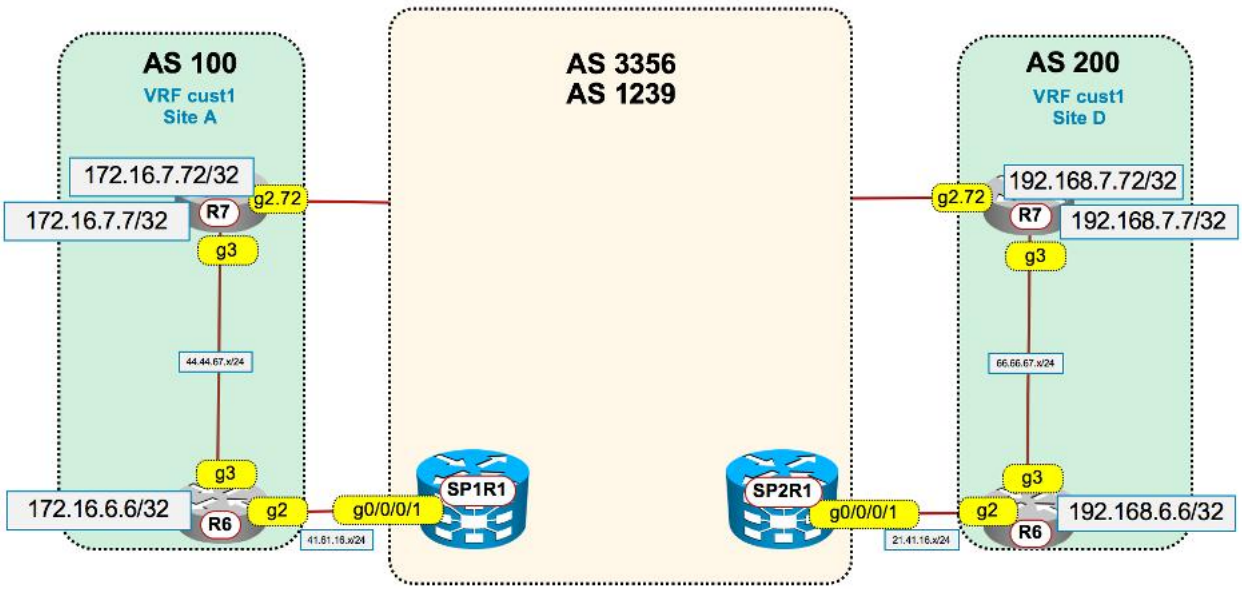
```
RP/0/0/CPU0:SP2R1(config)#router bgp 1239
RP/0/0/CPU0:SP2R1(config-bgp)#vrf cust1
RP/0/0/CPU0:SP2R1(config-bgp-vrf)#neighbor 21.41.16.6
RP/0/0/CPU0:SP2R1(config-bgp-vrf-nbr)#no shut

RP/0/0/CPU0:SP2R2(config)#router bgp 1239
RP/0/0/CPU0:SP2R2(config-bgp)#vrf cust1
RP/0/0/CPU0:SP2R2(config-bgp-vrf)#neighbor 20.40.27.7
RP/0/0/CPU0:SP2R2(config-bgp-vrf-nbr)#no shut
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 10:
MPLS L2VPN: E-LINE

In this lab you will build the L2VPN services provided by AS 100 and AS 200. Other service providers AS 3356 and AS 1239 play a role of CSC for L2VPN which means they offer a continuous LSP across Site A and Site D. AS 3356 and AS 1239 are unaware of L2VPN services.



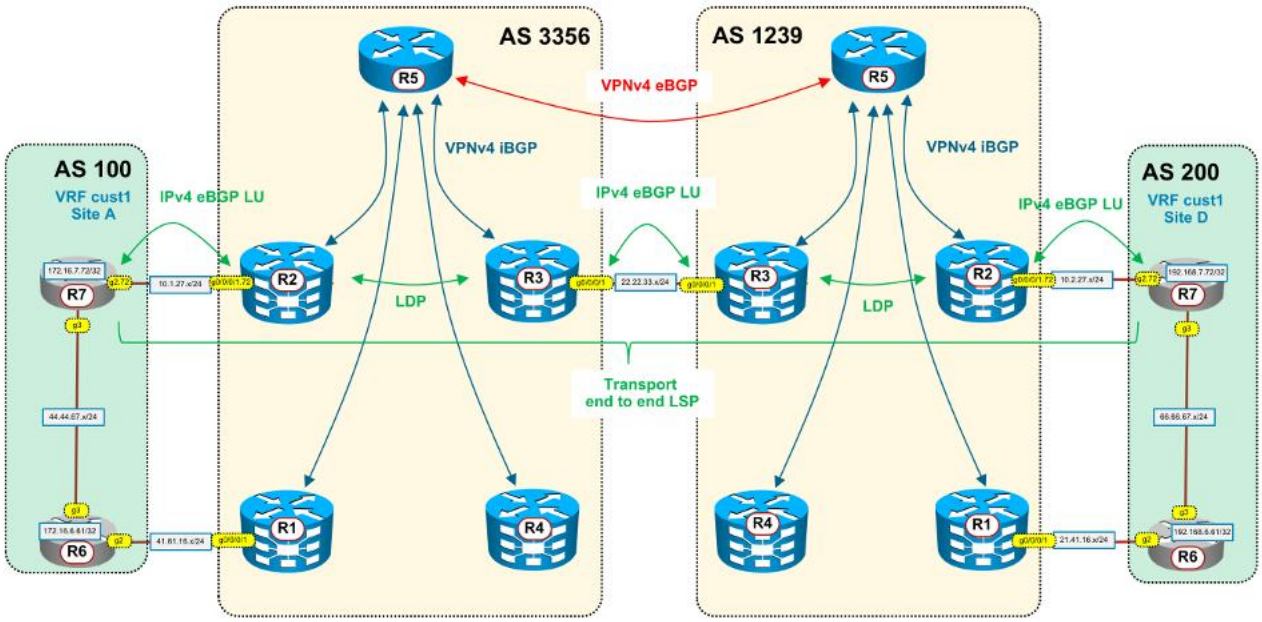
This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch10.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch10.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch10.unl`

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 0: Lab preparations

Task 0.1: Continuous LSP



In this task make sure that the LSP between CEL7 and CER7 is not broken and the path is taken correctly via R2 and the G2.72 subinterface on R7 routers.

Verification of the LSP.

```
CEL7#trace 192.168.7.72 sou 172.16.7.72
Type escape sequence to abort.
Tracing the route to 192.168.7.72
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.27.2 [MPLS: Label 102020 Exp 0] 27 msec 23 msec 30 msec
  2 33.56.23.3 [MPLS: Labels 103003/123223 Exp 0] 32 msec 24 msec 26 msec
  3 22.22.33.254 [MPLS: Labels 123304/123223 Exp 0] 27 msec 23 msec 25 msec
  4 12.39.23.2 [MPLS: Labels 0/123223 Exp 0] 27 msec 26 msec 24 msec
  5 10.2.27.7 37 msec *  28 msec
```

If R2 routers do not receive x.x.7.72/32 prefixes through the 10.x.27.2 BGP session possibly a filtering does not work in the current XE release. You can change a route-map to a prefix-list as follows.

```
CEL7(config)#ip prefix-list route72 permit 172.16.7.72/32
CEL7(config)#router bgp 100
CEL7(config-router)#add ipv4
CEL7(config-router-af)#no nei 10.1.27.2 route-map route72 out
CEL7(config-router-af)#nei 10.1.27.2 prefix-list route72 out

CER7(config)#ip prefix-list route72 permit 192.168.7.72/32
CER7(config)#router bgp 200
CER7(config-router)#add ipv4
CER7(config-router-af)#no nei 10.2.27.2 route-map route72 out
CER7(config-router-af)#nei 10.2.27.2 prefix-list route72 out
```

Verification of the x.x.7.72/32 prefix advertisement.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CEL7#sh ip bgp 192.168.7.72
BGP routing table entry for 192.168.7.72/32, version 86
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     26          28
  Refresh Epoch 1
  3356 1239 200
    10.1.27.2 from 10.1.27.2 (33.56.2.2)
      Origin incomplete, localpref 100, valid, external, best
      mpls labels in/out nolabel/102020
      rx pathid: 0, tx pathid: 0x0

CER7#sh ip bgp 172.16.7.72
BGP routing table entry for 172.16.7.72/32, version 251215
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     1           3
  Refresh Epoch 1
  1239 3356 100
    10.2.27.2 from 10.2.27.2 (12.39.2.2)
      Origin incomplete, localpref 100, valid, external, best
      mpls labels in/out nolabel/123218
      rx pathid: 0, tx pathid: 0x0
```

In both cases the BGP next hop is pointing correctly to R2's G0/0/0/1.72 subinterface.

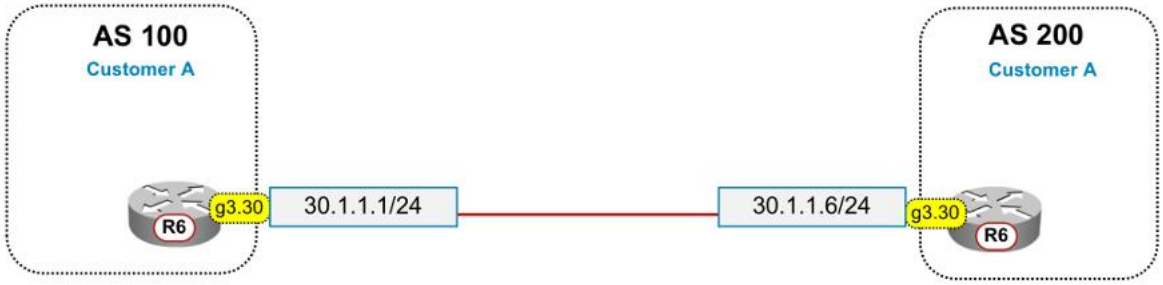
© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: VPWS

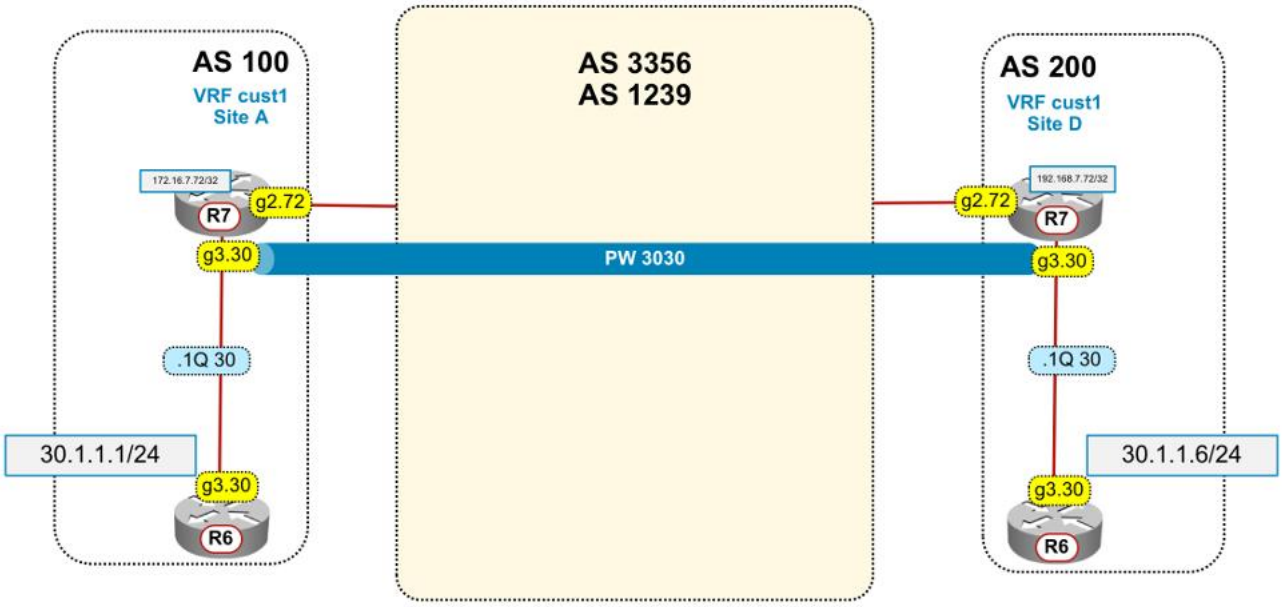
In this task you will configure a Virtual Private Wire Service (VPWS) which a L2 point-to-point service. You will utilize pseudowires (PW), VLAN tagging and other flexible and scalable methods to offer the L2VPN service.

Task 1.1: PW with .1Q

Customer A ordered a L2 point-to-point service to connect two sites. A logical topology is as depicted in the following figure.



The physical and logical topology from a standpoint of AS 100/200 is depicted as follows.



Add access subinterfaces.

```
CEL7(config)#int g3.30
CEL7(config-subif)#encapsulation dot1Q 30
CEL7(config-subif)#xconnect 192.168.7.72 3030 encapsulation mpls

CEL6(config)#int g3.30
CEL6(config-subif)#encapsulation dot1q 30
CEL6(config-subif)#ip add 30.1.1.1 255.255.255.0

CER7(config)#int g3.30
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CER7(config-subif)#encapsulation dot1q 30
CER7(config-subif)#xconnect 172.16.7.72 3030 encap mpls

CER6(config)#int g3.30
CER6(config-subif)#en d 30
CER6(config-subif)#ip add 30.1.1.6 255.255.255.0
```

Verification of connectivity.

```
CEL6#debug ip icmp
ICMP packet debugging is on

CEL6#ping 30.1.1.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.1.1.6, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

There is something wrong. Let's check the status of a pseudowire.

```
CEL7#sh mpls l2transport vc det
Local interface: Gi3.30 up, line protocol up, Eth VLAN 30 down
  Destination address: 192.168.7.72, VC ID: 3030, VC status: down
    Last error: Local access circuit is not ready for label advertise
    Output interface: none, imposed label stack {}
    Preferred path: not configured
    Default path: no route
    No adjacency
  Create time: 00:18:39, last status change time: 06:26:30
    Last label FSM state change time: 00:18:38
  Signaling protocol: LDP, peer unknown
    Targeted Hello: 172.16.7.7(LDP Id) -> 192.168.7.72, LDP is DOWN, no binding
    Graceful restart: not configured and not enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/None (no remote binding
      LDP route watch                   : enabled
      Label/status state machine        : local standby, AC-ready, LnuRnd
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: Not sent
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: No status
      Last remote LDP TLV    status rcvd: None (no remote binding)
      Last remote LDP ADJ    status rcvd: None (no remote binding)
    MPLS VC labels: local 38, remote unassigned
    Group ID: local 8, remote unknown
    MTU: local 1500, remote unknown
    Remote interface description:
  Sequencing: receive disabled, send disabled
  Control Word: On (configured: autosense)
  SSO Descriptor: 192.168.7.72/3030, local label: 38
  Dataplane:
    SSM segment/switch IDs: 0/0 (used), PWID: 1
  VC statistics:
    transit packet totals: receive 0, send 0
    transit byte totals:   receive 0, send 0
    transit packet drops:  receive 0, seq error 0, send 0
```

The pseudowire is down so let's check an LDP session with a neighbor.

```
CEL7#sh mpls ldp disc
 Local LDP Identifier:
    172.16.7.7:0
    Discovery Sources:
    Interfaces:
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
        GigabitEthernet3 (ldp): xmit/recv
            LDP Id: 172.16.6.6:0
    Targeted Hellos:
        172.16.7.7 -> 192.168.7.72 (ldp): active, xmit
```

CEL7 transmits targeted LDP Hellos but does not receive anything back from CER7.

By default routers do not allows to set up a targeted LDP session. To respond to requests for targeted LDP Hello messages add the following command.

```
CEL7(config)#mpls ldp discovery targeted-hello accept

CER7(config)#mpls ldp discovery targeted-hello accept
```

If your LDP session between CEL7 and CER7 is down make sure that there is a route to the LDP ID of a peer.

```
CEL7#sh mpls ldp disc
 Local LDP Identifier:
    172.16.7.7:0
    Discovery Sources:
    Interfaces:
        GigabitEthernet3 (ldp): xmit/recv
            LDP Id: 172.16.6.6:0
    Targeted Hellos:
        172.16.7.7 -> 192.168.7.72 (ldp): active, xmit/recv
            LDP Id: 192.168.7.7:0; no route
        172.16.7.72 -> 192.168.7.7 (ldp):  passive, xmit/recv
            LDP Id: 192.168.7.7:0; no route
```

Once you fix it the LDP session and pseudowire should be up.

```
CEL7#sh mpls ldp nei
    Peer LDP Ident: 172.16.6.6:0; Local LDP Ident 172.16.7.7:0
        TCP connection: 172.16.6.6.646 - 172.16.7.7.19759
        State: Oper; Msgs sent/rcvd: 678/572; Downstream
        Up time: 06:20:04
        LDP discovery sources:
          GigabitEthernet3, Src IP addr: 44.44.67.6
        Addresses bound to peer LDP Ident:
          41.61.16.6      44.44.67.6      21.41.46.6      172.16.6.6
          172.16.6.61     30.1.1.1
    Peer LDP Ident: 192.168.7.7:0; Local LDP Ident 172.16.7.7:0
        TCP connection: 192.168.7.7.20252 - 172.16.7.7.646
        State: Oper; Msgs sent/rcvd: 35/14; Downstream
        Up time: 00:00:19
        LDP discovery sources:
          Targeted Hello 172.16.7.72 -> 192.168.7.7, passive
          Targeted Hello 172.16.7.7 -> 192.168.7.72, active
        Addresses bound to peer LDP Ident:
          20.40.27.7      10.2.27.7       66.66.67.7      42.62.37.7
          192.168.7.7     192.168.7.71    192.168.7.72    192.168.7.73
```

You should see LDP Hellos transmitted (xmit) and received (recv).

```
CEL7#sh mpls ldp disc
 Local LDP Identifier:
    172.16.7.7:0
    Discovery Sources:
    Interfaces:
        GigabitEthernet3 (ldp): xmit/recv
            LDP Id: 172.16.6.6:0
    Targeted Hellos:
        172.16.7.7 -> 192.168.7.72 (ldp): active, xmit/recv
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
            LDP Id: 192.168.7.7:0
        172.16.7.72 -> 192.168.7.7 (ldp):  passive, xmit/recv
            LDP Id: 192.168.7.7:0
```

You can verify status of a pseudowire, a received VC (Virtual Circuit) label and the stack of labels to be imposed at the outgoing core interface which is Gi2.72 with two labels{102020 22}. The transport label is 102020 and the VC label is 22.

```
CEL7#sh mpls l2transport vc det
Local interface: Gi3.30 up, line protocol up, Eth VLAN 30 up
  Interworking type is Ethernet
  Destination address: 192.168.7.72, VC ID: 3030, VC status: up
    Output interface: Gi2.72, imposed label stack {102020 22}
    Preferred path: not configured
    Default path: active
    Next hop: 10.1.27.2
  Create time: 01:22:54, last status change time: 00:06:05
    Last label FSM state change time: 00:06:05
  Signaling protocol: LDP, peer 192.168.7.7:0 up
    Targeted Hello: 172.16.7.7(LDP Id) -> 192.168.7.72, LDP is UP
    Graceful restart: not configured and not enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 38, remote 22
    Group ID: local 8, remote 8
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  Control Word: On (configured: autosense)
  SSO Descriptor: 192.168.7.72/3030, local label: 38
  Dataplane:
    SSM segment/switch IDs: 4097/4096 (used), PWID: 1
  VC statistics:
    transit packet totals: receive 84, send 5
    transit byte totals:   receive 8546, send 666
    transit packet drops:  receive 0, seq error 0, send 0
```

Test a pseudowire.

```
CEL7#ping mpls pseudowire 192.168.7.72 3030
Sending 5, 72-byte MPLS Echos to 192.168.7.72,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labelled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'l' - Label switched with FEC change, 'd' - see DDMAP for return code,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
!!!!!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 23/33/70 ms
 Total Time Elapsed 182 ms
```

On IOS XE the 'debug ip icmp' command shows an output for locally generated packets only. Turn on the Embedded Packet Capture feature and test an end to end reachability.

```
CER6#monitor capture cap match any
CER6#monitor capture cap interface g3.30 both
CER6#monitor capture cap start
Started capture point : cap
CER6#
*Jan 01 19:39:12.053: %BUFCAP-6-ENABLE: Capture Point cap enabled.

CEL6#debug ip icmp
CEL6#ping 30.1.1.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.1.1.6, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 32/61/124 ms
CEL6#
*Jan 01 19:22:16.913: ICMP: echo reply rcvd, src 30.1.1.6, dst 30.1.1.1, topology BASE,
dscp 0 topoid 0
*Jan 01 19:22:16.953: ICMP: echo reply rcvd, src 30.1.1.6, dst 30.1.1.1, topology BASE,
dscp 0 topoid 0
*Jan 01 19:22:16.985: ICMP: echo reply rcvd, src 30.1.1.6, dst 30.1.1.1, topology BASE,
dscp 0 topoid 0
*Jan 01 19:22:17.035: ICMP: echo reply rcvd, src 30.1.1.6, dst 30.1.1.1, topology BASE,
dscp 0 topoid 0

CER6#monitor capture cap stop
Stopped capture point : cap
CER6#
*Jan 01 19:39:35.699: %BUFCAP-6-DISABLE: Capture Point cap disabled.
```

Examine captured packets.

```
CER6#show monitor capture cap buffer
 buffer size (KB) : 10240
 buffer used (KB) : 128
 packets in buf   : 15
 packets dropped  : 0
 packets per sec  : 0

CER6#show monitor capture cap buffer brief
 ----------------------------------------------------------------------------
 #   size   timestamp     source              destination      dscp    protocol
 ----------------------------------------------------------------------------
   0   78    0.000000   30.1.1.6         ->  224.0.0.10       48 CS6  EIGRP
   1   78    4.632022   30.1.1.6         ->  224.0.0.10       48 CS6  EIGRP
   2   78    9.424025   30.1.1.6         ->  224.0.0.10       48 CS6  EIGRP
   3   78   14.231982   30.1.1.6         ->  224.0.0.10       48 CS6  EIGRP
   4  118   15.859000   30.1.1.1         ->  30.1.1.6         0  BE   ICMP
   5  118   15.860007   30.1.1.6         ->  30.1.1.1         0  BE   ICMP
   6  118   15.886007   30.1.1.1         ->  30.1.1.6         0  BE   ICMP
   7  118   15.886007   30.1.1.6         ->  30.1.1.1         0  BE   ICMP
   8  118   15.912007   30.1.1.1         ->  30.1.1.6         0  BE   ICMP
   9  118   15.912998   30.1.1.6         ->  30.1.1.1         0  BE   ICMP
  10  118   15.938998   30.1.1.1         ->  30.1.1.6         0  BE   ICMP
  11  118   15.938998   30.1.1.6         ->  30.1.1.1         0  BE   ICMP
  12  118   15.966005   30.1.1.1         ->  30.1.1.6         0  BE   ICMP
  13  118   15.966005   30.1.1.6         ->  30.1.1.1         0  BE   ICMP
  14   78   18.855003   30.1.1.6         ->  224.0.0.10       48 CS6  EIGRP
```

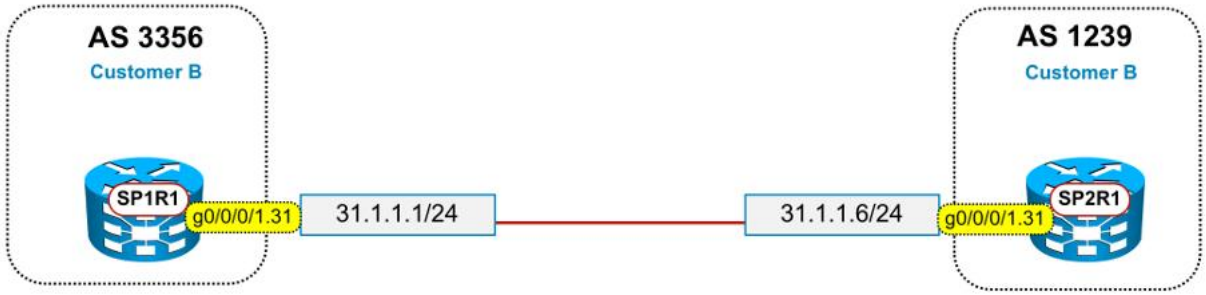

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

As you can see ICMP packets are received and replied by CER6 which completes this task.

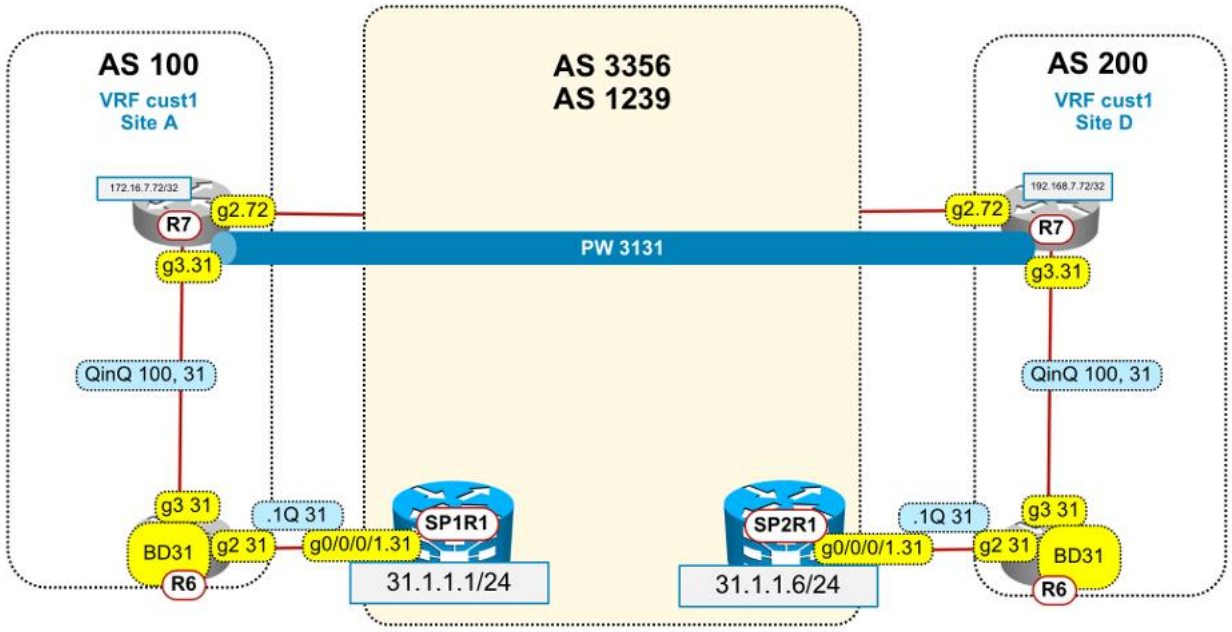
© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: PW with QinQ + local bridging

A logical topology of a L2 service for Customer B.



A physical/logical topology from a perspective of AS100/200.



Configure new .1Q subinterfaces using also service instances which is the EVC (Ethernet Virtual Circuit) syntax. Between R6 and R7 routers there is QinQ tagging to separate different customers in the access layer. VLAN tags are automatically removed and added on a subinterface (e.g. int G3.3 on CEL7). VLAN tags are not manipulated by default on EFP (Ethernet Flow Point) which is a service instance. The separate command 'rewrite' can be used to do so.

```
CEL6(config)#int g3
CEL6(config-if)#no ip add
CEL6(config-if)#no ipv6 add
CEL6(config-if)#service instance 31 ethernet
CEL6(config-if-srv)#encapsulation dot1q 100 second-dot1q 31
CEL6(config-if-srv)#rewrite ingress tag pop 1 symmetric
CEL6(config-if-srv)#bridge-domain 31
CEL6(config-if-srv)#int g2
CEL6(config-if)#no ip add
CEL6(config-if)#no ipv6 add
CEL6(config-if)#service instance 31 ethernet
CEL6(config-if-srv)#encapsulation dot1q 31
CEL6(config-if-srv)#bridge-domain 31
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CER6(config)#int g3
CER6(config-if)#no ip add
CER6(config-if)#no ipv6 add
CER6(config-if)#service instance 31 ethernet
CER6(config-if-srv)#encapsulation dot1q 100 second-dot1q 31
CER6(config-if-srv)# rewrite ingress tag pop 1 symmetric
CER6(config-if-srv)#bridge-domain 31
CER6(config-if-srv)#int g2
CER6(config-if)#no ip add
CER6(config-if)#no ipv6 add
CER6(config-if)#service instance 31 ethernet
CER6(config-if-srv)#encapsulation dot1q 31
CER6(config-if-srv)#bridge-domain 31

RP/0/0/CPU0:SP1R1(config)#int g0/0/0/1.31
RP/0/0/CPU0:SP1R1(config-subif)#encapsulation dot1q 31
RP/0/0/CPU0:SP1R1(config-subif)#ipv4 add 31.1.1.1/24

RP/0/0/CPU0:SP2R1(config)#int g0/0/0/1.31
RP/0/0/CPU0:SP2R1(config-subif)#encapsulation dot1q 31
RP/0/0/CPU0:SP2R1(config-subif)#ipv4 add 31.1.1.6/24
```

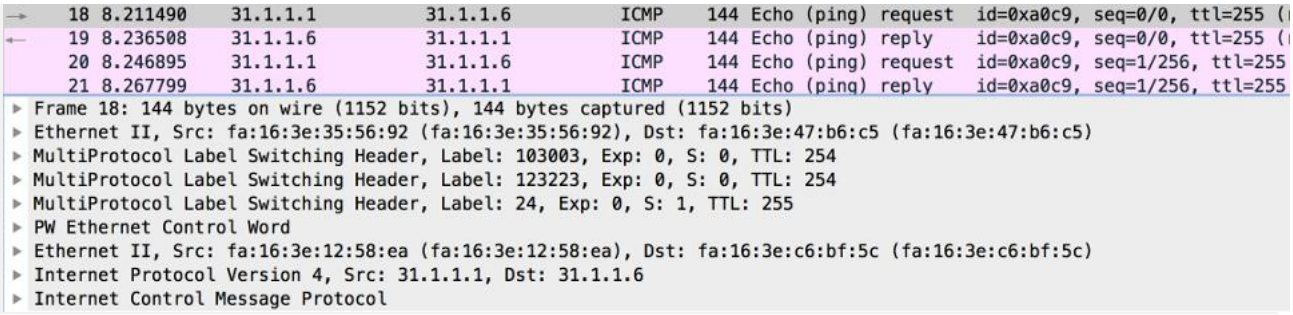
Attach a pseudowire to access ports on the MPLS edge routers.

```
CEL7(config)#int g3.31
CEL7(config-subif)#encapsulation dot1Q 100 second-dot1q 31
CEL7(config-subif)#xconnect 192.168.7.72 3131 encapsulation mpls

CER7(config)#int g3.31
CER7(config-subif)#encapsulation dot1Q 100 second-dot1q 31
CER7(config-subif)#xconnect 172.16.7.72 3131 encapsulation mpls

RP/0/0/CPU0:SP1R1#ping 31.1.1.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 31.1.1.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 29/29/29 ms
```

A packet capture at R2's G0/0/0/0 interface. As you can see there are no .1Q tags. They were removed at the ingress port of CEL7 router.



In the packet capture the transport label is 103003, the VPN label is 123223, the VC label is 24.

Verify the VPN label.

```
RP/0/0/CPU0:SP1R2#sh cef vrf csc 192.168.7.72
192.168.7.72/32, version 35, internal 0x1000001 0x0 (ptr 0xa14618f4) [1], 0x0 (0x0),
0x208 (0xa14f92f8)

 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 12.39.2.2/32, 5 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa1561b74 0x0]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 12.39.2.2/32 via 102008/0/21
     local label 102020
     next hop 33.56.23.3/32 Gi0/0/0/0     labels imposed {103003 123223}
```

The VPN label is received from SP2R2.

```
RP/0/0/CPU0:SP1R2#sh bgp vpnv4 u rd 72:72 labels | i "192.168.7.72|Local"
Network            Next Hop        Rcvd Label      Local Label
*>i192.168.7.72/32    12.39.2.2       123223          102020
```

As a result, the transport label from CEL7 102020 is swapped to the VPN label 123223 + the transport label 103003 is pushed.

```
RP/0/0/CPU0:SP1R2#sh mpls for labels 102020
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
102020 123223      192.168.7.72/32[V]              12.39.2.2       128771
```

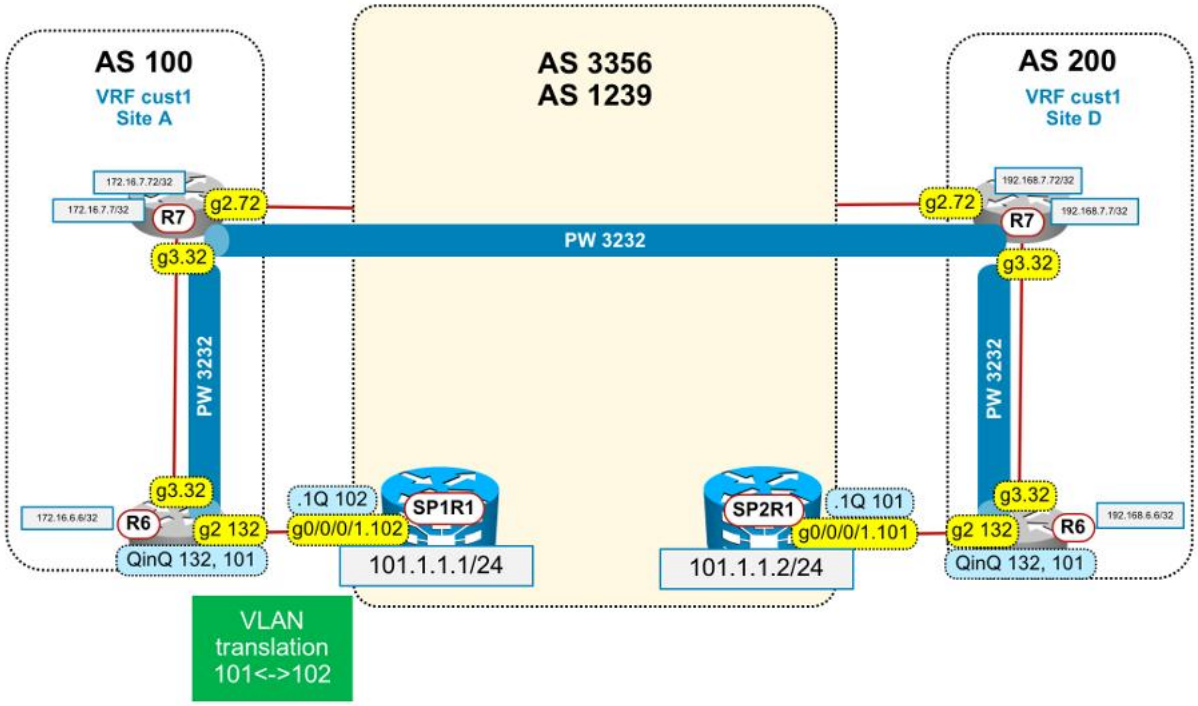
The most inner label 24 in the capture is the VC label. This is received from CER7.

```
CEL7#sh mpls l2transport vc 3131 det
Local interface: Gi3.31 up, line protocol up, Eth VLAN 100/31 up
  Interworking type is Ethernet
  Destination address: 192.168.7.72, VC ID: 3131, VC status: up
    Output interface: Gi2.72, imposed label stack {102020 24}
    Preferred path: not configured
    Default path: active
    Next hop: 10.1.27.2
  Create time: 00:15:30, last status change time: 00:14:44
    Last label FSM state change time: 00:14:44
  Signaling protocol: LDP, peer 192.168.7.7:0 up
    Targeted Hello: 172.16.7.7(LDP Id) -> 192.168.7.72, LDP is UP
    Graceful restart: not configured and not enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 16, remote 24
    Group ID: local 8, remote 8
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  Control Word: On (configured: autosense)
  SSO Descriptor: 192.168.7.72/3131, local label: 16
  Dataplane:
    SSM segment/switch IDs: 12292/8195 (used), PWID: 2
  VC statistics:
    transit packet totals: receive 16, send 36
    transit byte totals:   receive 2168, send 3672
    transit packet drops:  receive 0, seq error 0, send 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3: Multisegment PW with a VLAN tag translation

In this task you will configure a multisegment pseudowire with QinQ an a VLAN tag translation. A topology of this L2 service is as follows.



The multisegment pseudowire is set from 3 parts. Configure a new MPLS enabled link between R7 and R6 to provide an LSP for a pseudowire spanned in the access network.

```
CEL7(config)#int g3.32
CEL7(config-subif)#encapsulation dot1Q 32
CEL7(config-subif)#ip add 10.1.67.7 255.255.255.0
CEL7(config-subif)#mpls ip
CEL7(config-subif)#router ospf 100
CEL7(config-router)#net 10.1.67.0 0.0.0.255 a 0

CEL6(config)#int g3.32
CEL6(config-subif)#encapsulation dot1Q 32
CEL6(config-subif)#ip add 10.1.67.6 255.255.255.0
CEL6(config-subif)#mpls ip
CEL6(config-subif)#router ospf 100
CEL6(config-router)#net 10.1.67.0 0.0.0.255 a 0
```

Verify a new LDP session which should be established between R7 and R6.

```
CEL6#sh mpls ldp nei
    Peer LDP Ident: 172.16.7.7:0; Local LDP Ident 172.16.6.6:0
        TCP connection: 172.16.7.7.34143 - 172.16.6.6.646
        State: Oper; Msgs sent/rcvd: 32/29; Downstream
        Up time: 00:00:10
        LDP discovery sources:
          GigabitEthernet3.32, Src IP addr: 10.1.67.7
        Addresses bound to peer LDP Ident:
          40.60.27.7      10.1.27.7       44.44.67.7      22.42.37.7
          172.16.7.7      172.16.7.71     172.16.7.72     172.16.7.73
          10.1.67.7
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Configure a new pair of .1Q subinterfaces between R6 and a customer. Please note that the customer is using different VLAN IDs. The left site connected to AS 100 uses VLAN ID 102 and the right site connected to AS 200 uses VLAN ID 101. This is a mismatch and must be fixed by a VLAN translation. Otherwise the customer won't be able to communicate across sites.

```
CEL6(config)#int g2
CEL6(config-if)#service instance 132 ethernet
CEL6(config-if-srv)#encap dot1q 102
CEL6(config-if-srv)#rewrite ingress tag translate 1-to-2 dot1q 132 second-dot1q 101
symmetric

RP/0/0/CPU0:SP1R1(config)#interface GigabitEthernet0/0/0/1.102
RP/0/0/CPU0:SP1R1(config-subif)# ipv4 address 101.1.1.1 255.255.255.0
RP/0/0/CPU0:SP1R1(config-subif)# encapsulation dot1q 102
```

A similar configuration in AS 200 but without the VLAN translation.

```
CER7(config)# interface GigabitEthernet3.32
CER7(config-subif)# encapsulation dot1Q 32
CER7(config-subif)# ip address 10.2.67.7 255.255.255.0
CER7(config-subif)# mpls ip
CER7(config-subif)# router eigrp 100
CER7(config-router)#net 10.2.67.7 0.0.0.0

CER6(config)#int g3.32
CER6(config-subif)#encap dot1q 32
CER6(config-subif)#ip address 10.2.67.6 255.255.255.0
CER6(config-subif)#mpls ip
CER6(config)#interface GigabitEthernet2
CER6(config-if)#service instance 132 ethernet
CER6(config-if-srv)#encapsulation dot1q 101
CER6(config-if-srv)#rewrite ingress tag push dot1q 132 symmetric

RP/0/0/CPU0:SP2R1(config)#interface GigabitEthernet0/0/0/1.101
RP/0/0/CPU0:SP2R1(config-subif)# ipv4 address 101.1.1.2 255.255.255.0
RP/0/0/CPU0:SP2R1(config-subif)# encapsulation dot1q 101
```

Add and merge pseudowires to build one point-to-point wire.

```
CEL7(config)#l2 vfi mpw point-to-point
CEL7(config-vfi)#nei 192.168.7.72 3232 en mpls
CEL7(config-vfi)#nei 172.16.6.6 3232 en mpls

CEL6(config)#int g2
CEL6(config-if)#service instance 132 ethernet
CEL6(config-if-srv)#xconnect 172.16.7.7 3232 encapsulation mpls

CER7(config)#l2 vfi mpw point-to-point
CER7(config-vfi)#nei 172.16.7.72 3232 en mpls
CER7(config-vfi)#nei 192.168.6.6 3232 en mpls

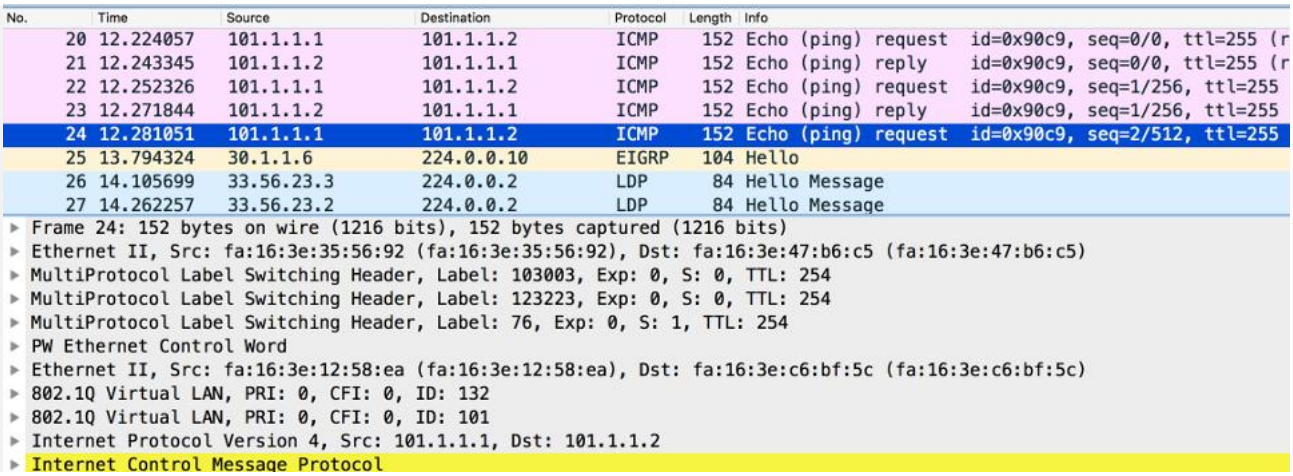
CER6(config)#int g2
CER6(config-if)#service instance 132 ethernet
CER6(config-if-srv)#xconnect 192.168.7.7 3232 encapsulation mpls
```

Verification of IP reachability.

```
RP/0/0/CPU0:SP1R1#ping 101.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 101.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 19/24/29 ms
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

A packet capture on SP1R2's G0/0/0/0 interface. You can see double VLAN tagging.



```
No.   Time        Source       Destination  Protocol Length Info
   20 12.224057   101.1.1.1    101.1.1.2    ICMP      152 Echo (ping) request  id=0x90c9, seq=0/0, ttl=255 (r
   21 12.243345   101.1.1.2    101.1.1.1    ICMP      152 Echo (ping) reply    id=0x90c9, seq=0/0, ttl=255 (r
   22 12.252326   101.1.1.1    101.1.1.2    ICMP      152 Echo (ping) request  id=0x90c9, seq=1/256, ttl=255
   23 12.271844   101.1.1.2    101.1.1.1    ICMP      152 Echo (ping) reply    id=0x90c9, seq=1/256, ttl=255
   24 12.281051   101.1.1.1    101.1.1.2    ICMP      152 Echo (ping) request  id=0x90c9, seq=2/512, ttl=255
   25 13.794324   30.1.1.6     224.0.0.10   EIGRP     104 Hello
   26 14.105699   33.56.23.3   224.0.0.2    LDP        84 Hello Message
   27 14.262257   33.56.23.2   224.0.0.2    LDP        84 Hello Message
▸ Frame 24: 152 bytes on wire (1216 bits), 152 bytes captured (1216 bits)
▸ Ethernet II, Src: fa:16:3e:35:56:92 (fa:16:3e:35:56:92), Dst: fa:16:3e:47:b6:c5 (fa:16:3e:47:b6:c5)
▸ MultiProtocol Label Switching Header, Label: 103003, Exp: 0, S: 0, TTL: 254
▸ MultiProtocol Label Switching Header, Label: 123223, Exp: 0, S: 0, TTL: 254
▸ MultiProtocol Label Switching Header, Label: 76, Exp: 0, S: 1, TTL: 254
▸ PW Ethernet Control Word
▸ Ethernet II, Src: fa:16:3e:12:58:ea (fa:16:3e:12:58:ea), Dst: fa:16:3e:c6:bf:5c (fa:16:3e:c6:bf:5c)
▸ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 132
▸ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 101
▸ Internet Protocol Version 4, Src: 101.1.1.1, Dst: 101.1.1.2
▸ Internet Control Message Protocol
```

You can verify MAC addresses of the customer and compare them with the packet capture.

```
RP/0/0/CPU0:SP1R1#sh int g0/0/0/1.102 | i Hardw
Hardware is VLAN sub-interface(s), address is fa16.3e12.58ea

RP/0/0/CPU0:SP2R1#sh int g0/0/0/1.101 | i Hardw
Hardware is VLAN sub-interface(s), address is fa16.3ec6.bf5c
```

MAC addresses are inside the inner Ethernet header.

```
CEL6#sh ethernet service instance int g2
Id    Type    Interface                      State     CE-Vlans
31    Static  GigabitEthernet2               Up
132   Static  GigabitEthernet2               Up
```

There are two parts of a PW on CEL7.

```
CEL7#sh mpls l2transport vc

Local intf     Local circuit                Dest address    VC ID       Status
-------------  ---------------------------- --------------- ----------- ----------
pw100004       192.168.7.72 3232            172.16.6.6      3232        UP
Gi3.30         Eth VLAN 30                  192.168.7.72    3030        UP
Gi3.31         Eth VLAN 100/31              192.168.7.72    3131        UP
pw100005       172.16.6.6 3232              192.168.7.72    3232        UP
```

You can verify the imposed label stack. There is just one VC label to CEL6 as this router is one hop away and the transport label is signalled as Imp-Null.

```
CEL7#sh mpls l2transport vc 3232 det
Local interface: pw100004 192.168.7.72 3232 up
  Destination address: 172.16.6.6, VC ID: 3232, VC status: up
    Output interface: Gi3.32, imposed label stack {42}
    Preferred path: not configured
    Default path: active
    Next hop: 10.1.67.6
  Create time: 00:50:03, last status change time: 00:00:00
    Last label FSM state change time: 00:29:11
  Signaling protocol: LDP, peer 192.168.7.7:0 up
    Targeted Hello: 172.16.7.7(LDP Id) -> 172.16.6.6, LDP is UP
    Graceful restart: not configured and not enabled
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 90, remote 42
    Group ID: local 16387, remote 7
    MTU: local 1500, remote 1500
    Remote interface description: to SP1R1
  Sequencing: receive transparent, send transparent
  Sequencing resync disabled
  Control Word: On (configured: autosense)
  SSO Descriptor: 172.16.6.6/3232, local label: 90
  Dataplane:
    SSM segment/switch IDs: 24588/12295 (used), PWID: 4
  VC statistics:
    transit packet totals: receive 46, send 37
    transit byte totals:   receive 6920, send 5404
    transit packet drops:  receive 0, seq error 0, send 0

Local interface: pw100005 172.16.6.6 3232 up
  Destination address: 192.168.7.72, VC ID: 3232, VC status: up
    Output interface: Gi2.72, imposed label stack {102020 76}
    Preferred path: not configured
    Default path: active
    Next hop: 10.1.27.2
  Create time: 00:50:03, last status change time: 00:00:00
    Last label FSM state change time: 00:29:11
  Signaling protocol: LDP, peer 172.16.6.6:0 up
    Targeted Hello: 172.16.7.7(LDP Id) -> 192.168.7.72, LDP is UP
    Graceful restart: not configured and not enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 88, remote 76
    Group ID: local 16388, remote 16387
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive transparent, send transparent
  Sequencing resync disabled
  Control Word: On (configured: autosense)
  SSO Descriptor: 192.168.7.72/3232, local label: 88
  Dataplane:
    SSM segment/switch IDs: 20493/12295 (used), PWID: 3
  VC statistics:
    transit packet totals: receive 37, send 46
    transit byte totals:   receive 5404, send 6920
    transit packet drops:  receive 0, seq error 0, send 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The customer wants to run a dynamic routing protocol over a newly deployed virtual link. Configure an OSPF session between SP1R1 and SP2R1.

```
RP/0/0/CPU0:SP1R1(config)#router ospf AS3356
RP/0/0/CPU0:SP1R1(config-ospf)# area 1
RP/0/0/CPU0:SP1R1(config-ospf-ar)#  nssa
RP/0/0/CPU0:SP1R1(config-ospf-ar)#  interface GigabitEthernet0/0/0/1.102

RP/0/0/CPU0:SP2R1(config)#router ospf AS1239
RP/0/0/CPU0:SP2R1(config-ospf)# area 1
RP/0/0/CPU0:SP2R1(config-ospf-ar)#  nssa
RP/0/0/CPU0:SP2R1(config-ospf-ar)#  interface GigabitEthernet0/0/0/1.101
```

Verify the OSPF session and received prefixes.

```
RP/0/0/CPU0:SP2R1#sh ospf nei
[..]
Neighbors for OSPF AS1239

Neighbor ID     Pri   State           Dead Time   Address         Interface
33.56.1.1       1     FULL/DR         00:00:36    101.1.1.1
GigabitEthernet0/0/0/1.101
    Neighbor is up for 00:00:08

Total neighbor count: 1

RP/0/0/CPU0:SP2R1#sh route os

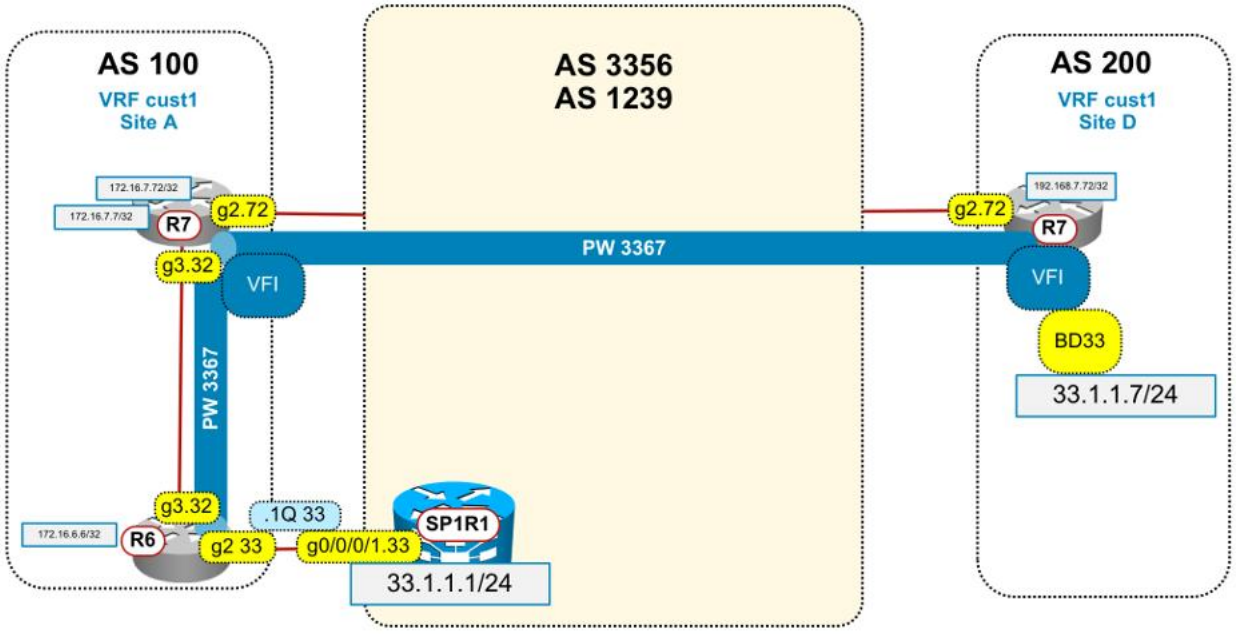
O IA 33.56.1.1/32 [110/2] via 101.1.1.1, 00:00:12, GigabitEthernet0/0/0/1.101
O IA 33.56.2.2/32 [110/3] via 101.1.1.1, 00:00:12, GigabitEthernet0/0/0/1.101
O IA 33.56.3.3/32 [110/3] via 101.1.1.1, 00:00:12, GigabitEthernet0/0/0/1.101
O IA 33.56.4.4/32 [110/3] via 101.1.1.1, 00:00:12, GigabitEthernet0/0/0/1.101
O IA 33.56.5.5/32 [110/3] via 101.1.1.1, 00:00:12, GigabitEthernet0/0/0/1.101
O IA 33.56.12.0/24 [110/2] via 101.1.1.1, 00:00:12, GigabitEthernet0/0/0/1.101
O IA 33.56.13.0/24 [110/2] via 101.1.1.1, 00:00:12, GigabitEthernet0/0/0/1.101
O IA 33.56.14.0/24 [110/2] via 101.1.1.1, 00:00:12, GigabitEthernet0/0/0/1.101
O IA 33.56.15.0/24 [110/2] via 101.1.1.1, 00:00:12, GigabitEthernet0/0/0/1.101
O IA 33.56.23.0/24 [110/3] via 101.1.1.1, 00:00:12, GigabitEthernet0/0/0/1.101
O IA 33.56.24.0/24 [110/3] via 101.1.1.1, 00:00:12, GigabitEthernet0/0/0/1.101
O IA 33.56.34.0/24 [110/3] via 101.1.1.1, 00:00:12, GigabitEthernet0/0/0/1.101
O IA 33.56.35.0/24 [110/3] via 101.1.1.1, 00:00:12, GigabitEthernet0/0/0/1.101
```

Will this path have an impact on the current L3VPN services? If so shut the service.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4: H-VPLS for E-LINE

A topology used in this task is depicted in the following figure.



To build E-LINE services you can use also typical multipoint solutions like VPLS. In this task you build a hierarchical VPLS service where the core between AS 100 and 200 run a VPLS instance and a spoke PW is attached to the VPLS in AS 100.

By default, in H-VPLS the traffic can be forwarded between pseudowires which are part of the VPLS (VFI) instance and spoke pseudowires which are connected to a bridge domain. In this task though the spoke pseudowire is connected to the VFI instance directly so the split horizon rule applies. To turn it off and allow the traffic to go between two pseudowires you can use the 'no-split-horizon' keyword.

Configure a new pair of .1Q subinterfaces, a bridge-domain with an assigned IP address in AS 100. Add a VPLS instance and pseudowires.

```
RP/0/0/CPU0:SP1R1(config)#interface GigabitEthernet0/0/0/1.33
RP/0/0/CPU0:SP1R1(config-subif)# ipv4 address 33.1.1.1 255.255.255.0
RP/0/0/CPU0:SP1R1(config-subif)# encapsulation dot1q 33

CEL6(config)#int g2
CEL6(config-if)#service instance 33 ethernet
CEL6(config-if-srv)#encapsulation dot1q 33
CEL6(config-if-srv)#rewrite ingress tag pop 1 symmetric
CEL6(config-if-srv)#xconnect 172.16.7.7 3367 encapsulation mpls

CEL7(config)#l2 vfi hvpls manual
CEL7(config-vfi)#vpn id 3367
CEL7(config-vfi)#bridge-domain 33
CEL7(config-vfi)#nei 172.16.6.6 3367 encapsulation mpls no-split-horizon
```

The first part of the VPLS solution is ready.

```
CEL7#sh mpls l2transport vc
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
pw100001       192.168.7.72 3232          172.16.6.6      3232       UP
VFI hvpls      vfi                        172.16.6.6      3367       UP
Gi3.30         Eth VLAN 30                192.168.7.72    3030       UP
Gi3.31         Eth VLAN 100/31            192.168.7.72    3131       UP
pw100002       172.16.6.6 3232            192.168.7.72    3232       UP

CEL6#sh mpls l2 vc

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
Gi2            Eth VLAN 102               172.16.7.7      3232       UP
Gi2            Eth VLAN 33                172.16.7.7      3367       UP
```

Add a bridge domain in AS 200 and a pseudowire between AS 100 and 200.

```
CEL7(config)#l2 vfi hvpls manual
CEL7(config-vfi)#nei 192.168.7.72 encapsulation mpls

CER7(config)#l2 vfi hvpls manual
CER7(config-vfi)# vpn id 3367
CER7(config-vfi)# bridge-domain 33
CER7(config-vfi)#nei 172.16.7.72 en mpls
```

The BDI interface is a routed representation of a bridge domain.

```
CER7(config)#int bdi 33
CER7(config-if)#ip add 33.1.1.7 255.255.255.0
CER7(config-if)#no shut
```

Verification of virtual circuits.

```
CEL6#sh mpls l2 vc

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
Gi2            Eth VLAN 102               172.16.7.7      3232       UP
Gi2            Eth VLAN 33                172.16.7.7      3367       UP

CEL7#sh mpls l2transport vc

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
pw100001       192.168.7.72 3232          172.16.6.6      3232       UP
VFI hvpls      vfi                        172.16.6.6      3367       UP
Gi3.30         Eth VLAN 30                192.168.7.72    3030       UP
Gi3.31         Eth VLAN 100/31            192.168.7.72    3131       UP
pw100002       172.16.6.6 3232            192.168.7.72    3232       UP
VFI hvpls      vfi                        192.168.7.72    3367       UP

CER7#sh mpls l2 vc 3367

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
VFI hvpls      vfi                        172.16.7.72     3367       UP
```

You can check the status of BD 33 and learnt MAC addresses.

```
CEL7#sh bridge-domain 33
Bridge-domain 33 (2 ports in all)
State: UP                    Mac learning: Enabled
Aging-Timer: 300 second(s)
    vfi hvpls neighbor 192.168.7.72 3367
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    vfi hvpls neighbor 172.16.6.6 3367
   AED MAC address    Policy  Tag       Age  Pseudoport
   0   FA16.3E12.58EA forward dynamic   64   hvpls.404014
   0   001E.E5F7.AFBF forward dynamic   64   hvpls.404013
```

Compare MAC addresses with source interfaces.

```
RP/0/0/CPU0:SP1R1#sh int g0/0/0/1.33 | i Hard
  Hardware is VLAN sub-interface(s), address is fa16.3e12.58ea

CER7#sh int bdi 33 | i Hard
  Hardware is BDI, address is 001e.e5f7.afbf (bia 001e.e5f7.afbf)

RP/0/0/CPU0:SP1R1#sh arp | i 33.1.1.
33.1.1.1        -          fa16.3e12.58ea  Interface  ARPA  GigabitEthernet0/0/0/1.33
33.1.1.7        02:37:42   001e.e5f7.afbf  Dynamic    ARPA  GigabitEthernet0/0/0/1.33
```

MAC addresses are learnt in the data plane correctly.

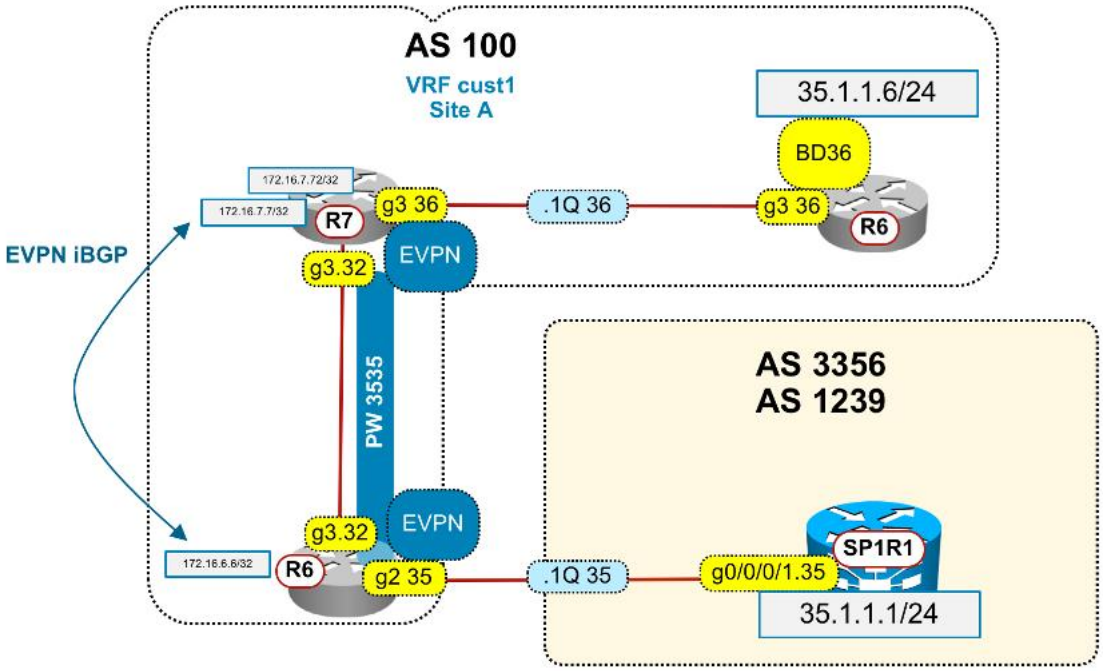
Verify the IP reachability.

```
RP/0/0/CPU0:SP1R1#ping 33.1.1.7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 33.1.1.7, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 19/27/29 ms
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.5: Point to point EVPN

In this task you will configure EVPN to offer the E-LINE service according to the following topology. Please note that R6 plays here a role of a PE and a CE at the same time. The traffic is isolated on R6 between two legs of the access layer by using VLAN IDs 35 and 36.



Configure new pairs of .1Q subinterfaces and a bridge domain.

```
RP/0/0/CPU0:SP1R1(config)#interface GigabitEthernet0/0/0/1.35
RP/0/0/CPU0:SP1R1(config-subif)# ipv4 address 35.1.1.1 255.255.255.0
RP/0/0/CPU0:SP1R1(config-subif)# encapsulation dot1q 35

CEL6(config)#int g2
CEL6(config-if)# service instance 35 ethernet
CEL6(config-if-srv)#  encapsulation dot1q 35
CEL6(config-if-srv)#  rewrite ingress tag pop 1 symmetric

CEL6(config)#int g3
CEL6(config-if)# service instance 36 ethernet
CEL6(config-if-srv)#  encapsulation dot1q 36
CEL6(config-if-srv)#  rewrite ingress tag pop 1 symmetric
CEL6(config-if-srv)#  bridge-domain 36
CEL6(config-if-srv)#int bdi 36
CEL6(config-if)#ip add 35.1.1.6 255.255.255.0
CEL6(config-if)#no shut

CEL7(config)#int g3
CEL7(config-if)#no ip add
CEL7(config-if)#no ipv6 add
CEL7(config-if)# service instance 36 ethernet
CEL7(config-if-srv)#  encapsulation dot1q 36
CEL7(config-if-srv)#  rewrite ingress tag pop 1 symmetric
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Configure the point-to-point EVPN instance.

```
CEL7(config)#l2vpn evpn instance 1 point-to-point
CEL7(config-evpn-evi)# vpws context evpn1
CEL7(config-evpn-vpws)#  service target 3535 source 3535
CEL7(config-evpn-vpws)#  member GigabitEthernet3 service-instance 36

CEL6(config)#l2vpn evpn instance 1 point-to-point
CEL6(config-evpn-evi)# vpws context evpn1
CEL6(config-evpn-vpws)#  service target 3535 source 3535
CEL6(config-evpn-vpws)#  member GigabitEthernet2 service-instance 35
```

Enable the EVPN address family to automatically advertise Ethernet segments.

```
CEL7(config)#router bgp 100
CEL7(config-router)# address-family l2vpn evpn
CEL7(config-router-af)#  neighbor 172.16.6.6 activate

CEL6(config)#router bgp 100
CEL6(config-router)# address-family l2vpn evpn
CEL6(config-router-af)#  neighbor 172.16.7.7 activate
```

A BGP session verification.

```
CEL7#show l2vpn evpn summary
L2VPN EVPN VPWS:
  EVIs (point-to-point): 1
  Total VCs: 1
    1 up, 0 down, 0 admin-down, 0 hot-standby, 0 other
  Total EVPN EAD routes: 2
    1 local, 1 remote
  Total EVI EAD routes: 2
    1 local, 1 remote (1 in-use)
  BGP: ASN 100, address-family l2vpn evpn configured
  Router ID: 172.16.7.7
```

Run the command to see the contents of the EVPN route. The Ethernet segment has its own unique Ethernet Tag 3535 to identify a potential traffic path.

```
CEL7#show l2vpn evpn rib ead
+- Origin of entry                                (i=iBGP/e=eBGP/L=Local)
| +- Best path                                    (Yes/No)?
| |
v v
O B         RD              Ethernet Segment Id    Eth Tag     Next Hop
-+-+--------------------+------------------------+--------+---------------
i Y 172.16.6.6:1          0000.0000.0000.0000.0000 3535     172.16.6.6
L - 172.16.7.7:1          0000.0000.0000.0000.0000 3535
```

To see an allocated label to R6, type the following command.

```
CEL7#show l2vpn evpn rib ead evi
+- Origin of entry                                (i=iBGP/e=eBGP/L=Local)
| +- Provisioned                                  (Yes/No)?
| | +- Best path                                  (Yes/No)?
| | |
v v v
O P B  EVI    Ethernet Segment Id    Eth Tag     Next Hop      Label
-+-+-+-----+------------------------+--------+---------------+--------
i Y Y 1     0000.0000.0000.0000.0000 3535     172.16.6.6      22
L - - 1     0000.0000.0000.0000.0000 3535                     26
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

RFC7432 states that "For a multihomed site, each Ethernet segment (ES) is identified by a unique non-zero identifier called an Ethernet Segment Identifier (ESI)". In this case ESI is zero as the site is single-homed.

In the EVPN table you can see that there are just type 1 of routes which are Ethernet Autodiscovery Routes to carry among others RD, ESI, ETag of the Ethernet segment.

```
CEL7#show bgp l2vpn evpn
BGP table version is 4, local router ID is 172.16.7.7
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 172.16.6.6:1
 *>i  [1][172.16.6.6:1][00000000000000000000][3535]/23
                       172.16.6.6               0    100      0 ?
Route Distinguisher: 172.16.7.7:1
 *>   [1][172.16.7.7:1][00000000000000000000][3535]/23
                       ::                                  32768 ?
```

Verify the status of a virtual circuit.

```
CEL7#show l2vpn evpn vc all
EVPN ID Source   Target   Type   Name/Interface                   Status
------- -------- -------- ------ -------------------------------- ----------
1       3535     3535     p2p    evpn1                            up
                                 Gi3:36                           up

CEL7#show l2vpn evpn vc all detail
EVPN name: evpn1, state: up, type: point-to-point
  EVPN ID: 1
  VPWS Service Instance ID: Source 3535, Target 3535
  Labels: Local 26, Remote 22
  Next Hop Address: 172.16.6.6
  Associated member interface Gi3 up, Gi3:36 status is up
  Dataplane:
    SSM segment/switch IDs: 54060/29483 (used), PWID: 49
  Rx Counters
    12 input transit packets, 1584 bytes
    0 drops
  Tx Counters
    11 output transit packets, 1476 bytes
    0 drops
  5 VC FSM state transitions, Last 5 shown
    Prov: Idle -> Prov, Mon Jan 01 23:39:05.140 (00:20:31 ago)
    AdmUp: Prov -> LocWait, Mon Jan 01 23:39:05.140 (00:20:31 ago)
    LocUp: LocWait -> RemWait, Mon Jan 01 23:39:05.142 (00:20:31 ago)
    RemUp: RemWait -> Act, Mon Jan 01 23:39:05.142 (00:20:31 ago)
    DpUp: Act -> Est, Mon Jan 01 23:39:05.150 (00:20:30 ago)
```

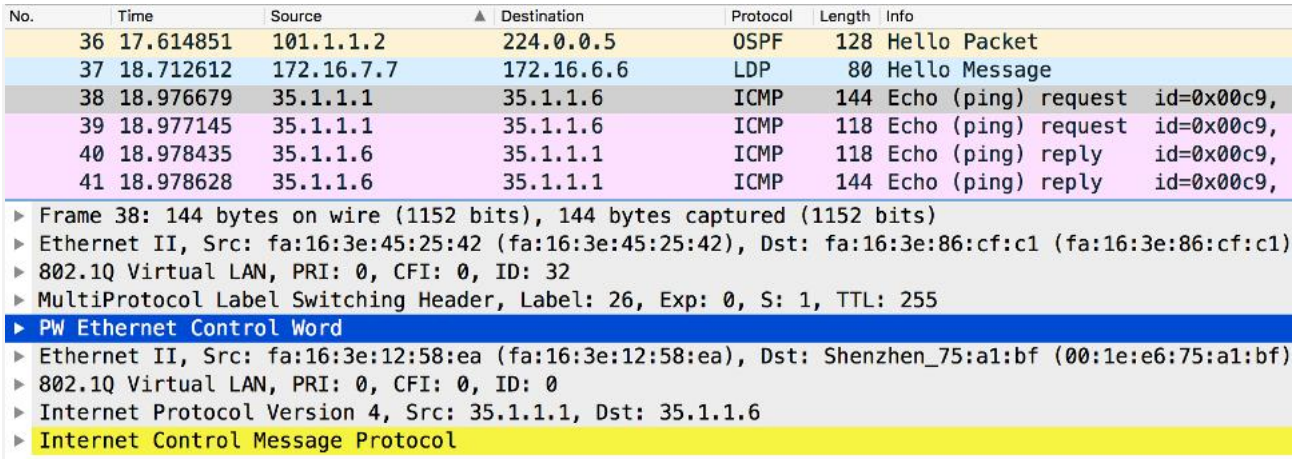
Verify the IP reachability.

```
RP/0/0/CPU0:SP1R1#ping 35.1.1.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 35.1.1.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/13/29 ms
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The following figure is a packet capture on CEL6's G2 interface. You can see a dummy VLAN tag with ID zero. If the Ethernet pseudowire is running in VLAN mode at the PE device, the PE device requires to receive the frame with a service-delimiting tag. In this task edge devices remove the tag at the ingress access port so a dummy tag is prepended in the frame before forwarding the frame over to Ethernet PW.

Do you remember why there is an external VLAN tag 32?



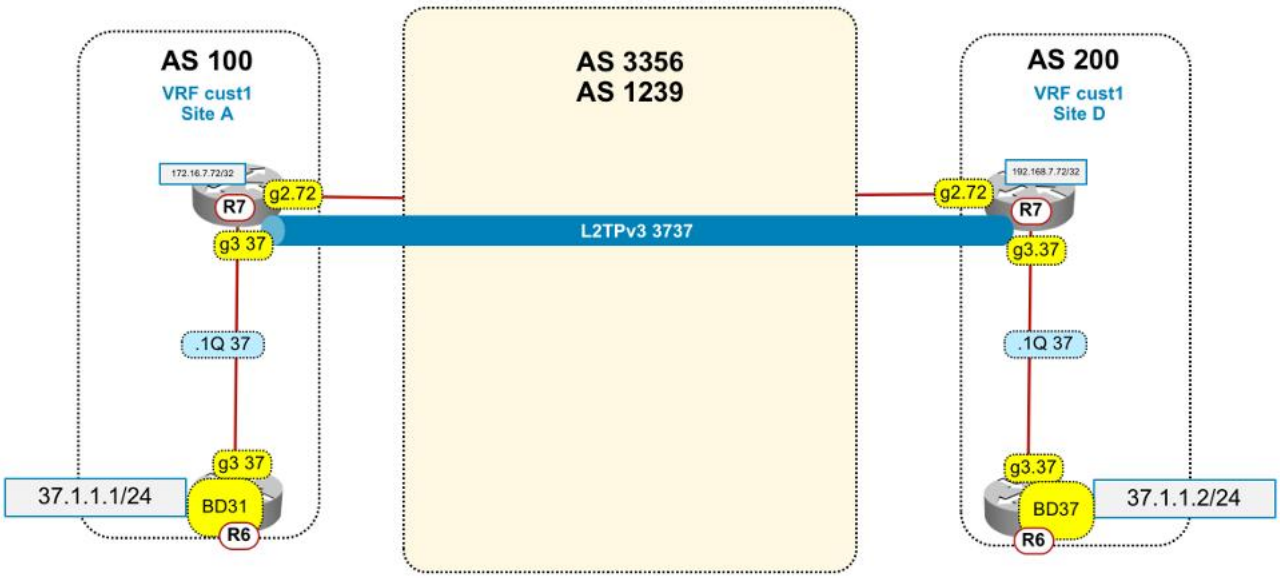
Verify MAC learning on a bridge domain and compare it with the packet capture.

```
CEL6#sh bridge-domain 36
Bridge-domain 36 (2 ports in all)
State: UP                    Mac learning: Enabled
Aging-Timer: 300 second(s)
    BDI36  (up)
    GigabitEthernet3 service instance 36
   AED MAC address    Policy  Tag       Age  Pseudoport
   0   FA16.3E12.58EA forward dynamic   138  GigabitEthernet3.EFP36
   -   001E.E675.A1BF to_bdi  static    0    BDI36
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.6: L2TPv3

Configure L2TPv3 protocol to offer the L2VPN service without a need of MPLS.



Configure new pairs of .1Q subinterfaces and bridge-domains to simulate a customer.

```
CEL6(config)#int g3
CEL6(config-if)#service instance 37 eth
CEL6(config-if-srv)#encapsulation dot1q 37
CEL6(config-if-srv)#rewrite ing ta pop 1 symmetric
CEL6(config-if-srv)#bridge-domain 37
CEL6(config-if-srv)#int bdi 37
CEL6(config-if)#ip add 37.1.1.1 255.255.255.0


CER6(config)#int g3.37
CER6(config-subif)#encap dot1q 37
CER6(config-subif)#ip add 37.1.1.2 255.255.255.0
```

Configure a PW class to parametrize L2TPv3 settings. Here you can configure an authentication of a circuit (not in this example) and a source interface. Attach the L2TPv3 tunnel to an Ethernet Flow Point 37.

```
CEL7(config)#pseudowire-class l2tpv3
CEL7(config-pw-class)#encap l2tpv3
CEL7(config-pw-class)#ip local inter lo2
CEL7(config-pw-class)#int g3
CEL7(config-if)#service instance 37 eth
CEL7(config-if-srv)#encap dot1q 37
CEL7(config-if-srv)#rewrite ingress tag pop 1 symmetric
CEL7(config-if-srv)#xconnect 192.168.7.72 3737 pw-class l2tpv3

CER7(config)#pseudowire-class l2tpv3
CER7(config-pw-class)# encapsulation l2tpv3
CER7(config-pw-class)# ip local interface Loopback2
CER7(config-pw-class)#int g3.37
CER7(config-subif)#encap dot1q 37
CER7(config-subif)#xconnect 172.16.7.72 3737 pw-class l2tpv3
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Verify the L2VPN service.

```
CEL7#sh l2vpn service name Gi3.37-37 det
Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down             AD=Admin Down     IA=Inactive
        SB=Standby  HS=Hot Standby      RV=Recovering     NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                   Prio  St  XC St
  ---------          -----       -------------                   ----  --  -----
VPWS name: Gi3.37-37, State: UP
  Gi3.37             left        Gi3.37:37(Eth VLAN)             0     UP  UP
                                 Interworking: vlan
  pw100007           right       192.168.7.72:3737(L2TP)         0     UP  UP
                                 Session ID: 440109028
                                 Tunnel ID: 3471408754
                                 Protocol State: UP
                                 Remote Circuit State: UP
                                 pw-class: l2tpv3
```

A detailed information on a far end.

```
CER7#sh l2vpn service name Gi3.37-37 det
Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down             AD=Admin Down     IA=Inactive
        SB=Standby  HS=Hot Standby      RV=Recovering     NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                   Prio  St  XC St
  ---------          -----       -------------                   ----  --  -----
VPWS name: Gi3.37-37, State: UP
  Gi3.37             left        Gi3.37:37(Eth VLAN)             0     UP  UP
                                 Interworking: vlan
  pw100012           right       172.16.7.72:3737(L2TP)          0     UP  UP
                                 Session ID: 1067367809
                                 Tunnel ID: 602870362
                                 Protocol State: UP
                                 Remote Circuit State: UP
                                 pw-class: l2tpv3
```

Check the IP reachability.

```
CEL6#ping 37.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 37.1.1.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 23/26/32 ms
CEL6#
*Jan 01 22:32:56.608: ICMP: echo reply rcvd, src 37.1.1.2, dst 37.1.1.1, topology BASE,
dscp 0 topoid 0
*Jan 01 22:32:56.635: ICMP: echo reply rcvd, src 37.1.1.2, dst 37.1.1.1, topology BASE,
dscp 0 topoid 0
*Jan 01 22:32:56.668: ICMP: echo reply rcvd, src 37.1.1.2, dst 37.1.1.1, topology BASE,
dscp 0 topoid 0
*Jan 01 22:32:56.692: ICMP: echo reply rcvd, src 37.1.1.2, dst 37.1.1.1, topology BASE,
dscp 0 topoid 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Verify learnt MAC addresses. In this example the neighbor 37.1.1.2 has the FA16.3E6A.4595 MAC address.

```
CEL6#sh bridge-domain 37
Bridge-domain 37 (2 ports in all)
State: UP                    Mac learning: Enabled
Aging-Timer: 300 second(s)
    BDI37  (up)
    GigabitEthernet3 service instance 37
   AED MAC address    Policy  Tag       Age  Pseudoport
   -   001E.E675.A1BF to_bdi  static    0    BDI37
   0   FA16.3E6A.4595 forward dynamic   296  GigabitEthernet3.EFP37

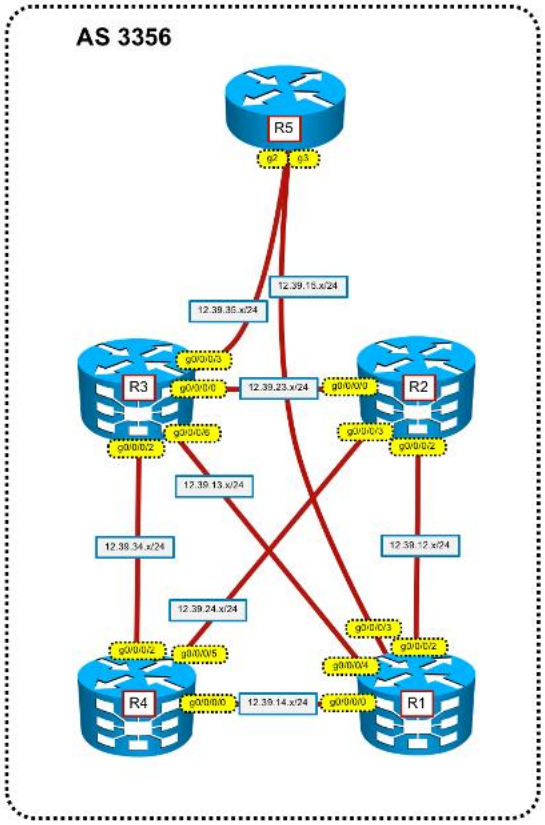
CEL6#sh ip arp 37.1.1.2
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  37.1.1.2                9   fa16.3e6a.4595  ARPA   BDI37
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 11:
MPLS L2VPN: E-LAN, E-TREE

In this lab you will configure E-LAN and E-TREE services. For E-LAN the following physical topology of AS 3356 is used:

IPv4
Infrastructure:
33.56.RxRy.x/24
Loopbacks:
33.56.Rx.Rx/32



A physical topology for E-TREE is the same as for E-LINE.

This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch11.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch11.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch11.unl`

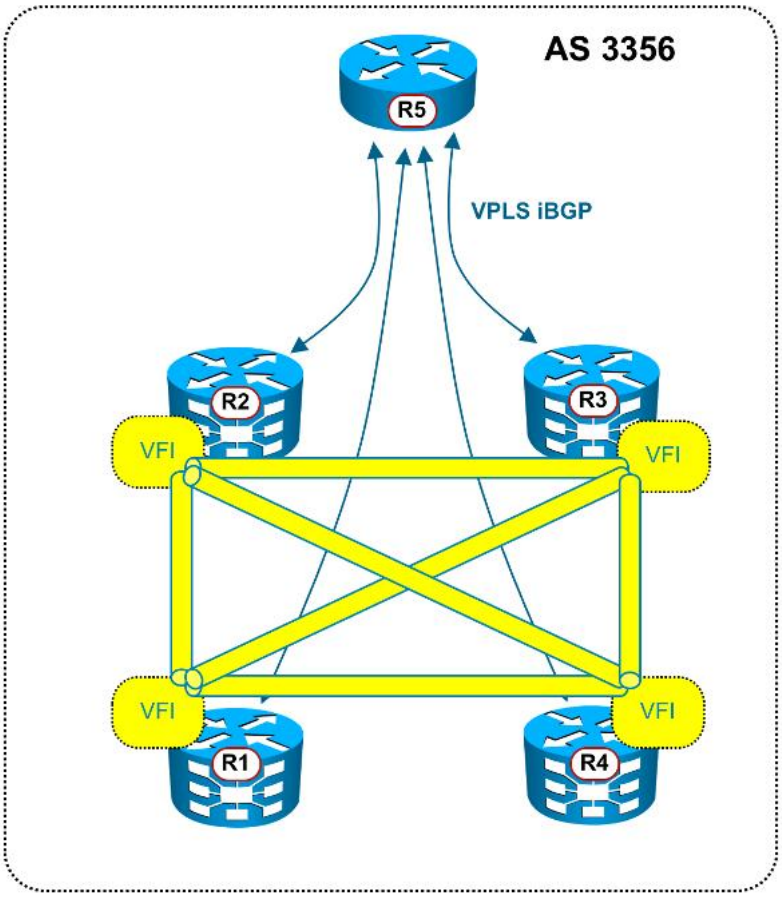
Please note that E-LAN services to be configured on IOS XR and an example of the BGP signalled VPLS on IOS XE do not offer a working data plane for a virtualized image. Therefore, you can configure and validate the control-plane only.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: VPLS

Task 1.1: E-LAN BGP signalled VPLS

Configure an E-LAN service as depicted on the following figure.



The first step is to enable the VPLS address family. You need to disable (suppress) LDP as a signalization protocol for VC labels.

```
SP1-R5-RR(config)#router bgp 3356
SP1-R5-RR(config-router)#add l2vpn vpls
SP1-R5-RR(config-router-af)#  neighbor 33.56.1.1 activate
SP1-R5-RR(config-router-af)#  neighbor 33.56.1.1 send-community extended
SP1-R5-RR(config-router-af)#  neighbor 33.56.1.1 route-reflector-client
SP1-R5-RR(config-router-af)#  neighbor 33.56.1.1 suppress-signaling-protocol ldp
SP1-R5-RR(config-router-af)#  neighbor 33.56.2.2 activate
SP1-R5-RR(config-router-af)#  neighbor 33.56.2.2 send-community extended
SP1-R5-RR(config-router-af)#  neighbor 33.56.2.2 route-reflector-client
SP1-R5-RR(config-router-af)#  neighbor 33.56.2.2 suppress-signaling-protocol ldp
SP1-R5-RR(config-router-af)#  neighbor 33.56.3.3 activate
SP1-R5-RR(config-router-af)#  neighbor 33.56.3.3 send-community extended
SP1-R5-RR(config-router-af)#  neighbor 33.56.3.3 route-reflector-client
SP1-R5-RR(config-router-af)#  neighbor 33.56.3.3 suppress-signaling-protocol ldp
SP1-R5-RR(config-router-af)#  neighbor 33.56.4.4 activate
SP1-R5-RR(config-router-af)#  neighbor 33.56.4.4 send-community extended
SP1-R5-RR(config-router-af)#  neighbor 33.56.4.4 route-reflector-client
SP1-R5-RR(config-router-af)#  neighbor 33.56.4.4 suppress-signaling-protocol ldp
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Add the following configuration on each of the route reflector client R1, R2, R3, R4.

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#add l2vpn vpls-vpws
RP/0/0/CPU0:SP1R2(config-bgp-af)#nei 33.56.5.5
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#add l2vpn vpls-vpws
```

Add on each RR client a new subinterface type l2transport (EFP on XR) attached to a bridge domain which is linked with the VPLS instance - VFI (Virtual Forwarding Instance). The VPLS instance is identified by VPN ID which should be unique globally. You can have one VPLS instance per each of a customer or a group of customers. VE ID must be unique among edge devices. Assign VE ID 1 for R1, 2 for R2, 3 for R3, 4 for R4. RR does not need the VPLS instance configuration as this is not the edge device.

```
RP/0/0/CPU0:SP1R2(config)#int g0/0/0/1.40 l2transport
RP/0/0/CPU0:SP1R2(config-subif)#encap dot1q 40
RP/0/0/CPU0:SP1R2(config-subif)#!
RP/0/0/CPU0:SP1R2(config-subif)#l2vpn
RP/0/0/CPU0:SP1R2(config-l2vpn)# bridge group bg1
RP/0/0/CPU0:SP1R2(config-l2vpn-bg)# bridge-domain bd1
RP/0/0/CPU0:SP1R2(config-l2vpn-bg-bd)#interface GigabitEthernet0/0/0/1.40
RP/0/0/CPU0:SP1R2(config-l2vpn-bg-bd-ac)#vfi vf1
RP/0/0/CPU0:SP1R2(config-l2vpn-bg-bd-vfi)#vpn-id 100
RP/0/0/CPU0:SP1R2(config-l2vpn-bg-bd-vfi)#autodiscovery bgp
RP/0/0/CPU0:SP1R2(config-l2vpn-bg-bd-vfi-ad)#rd 4:4
RP/0/0/CPU0:SP1R2(config-l2vpn-bg-bd-vfi-ad)#route-target 40:40
RP/0/0/CPU0:SP1R2(config-l2vpn-bg-bd-vfi-ad)#signaling-protocol bgp
RP/0/0/CPU0:SP1R2(config-l2vpn-bg-bd-vfi-ad-sig)#ve-id 2
```

Autodiscovery is managed by BGP so any new VPLS instances will be advertised automatically. BGP is also used to signal the VC label. You don't need to suppress LDP in IOS XR as it is required in IOS XE.

Verify a BGP session. There are 3 VPLS instances advertised.

```
RP/0/0/CPU0:SP1R2#sh bgp l2vpn vpls summ
[..]
Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
33.56.5.5         0  3356  212357    5785        8    0    0 00:00:23          3
```

As an advertised route there is a VPLS instance with RD, a next hop to show the path how to reach this instance, a VC label and other standard attributes.

```
RP/0/0/CPU0:SP1R2#sh bgp l2vpn vpls
BGP router identifier 33.56.2.2, local AS number 3356
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 8
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop        Rcvd Label      Local Label
Route Distinguisher: 4:4 (default for vrf bg1:bd1)
*>i1:1/32             33.56.1.1       101015          nolabel
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
*> 2:1/32              0.0.0.0          nolabel          102030
*>i3:1/32             33.56.3.3        103030           nolabel
*>i4:1/32             33.56.4.4        104075           nolabel

Processed 4 prefixes, 4 path
```

Check the status of a bridge domain.

```
RP/0/0/CPU0:SP1R2#sh l2vpn bridge-domain bd-name bd1 brief
Legend: pp = Partially Programmed.
Bridge Group:Bridge-Domain Name  ID    State          Num ACs/up    Num PWs/up    Num
PBBs/up Num VNIs/up
-------------------------------- ----- -------------- ------------ ------------- -------
---- -----------
bg1:bd1                          0     up             1/1           3/3           0/0
0/0
```

Examine detailed information about established PWs and attachment circuits (access interfaces). VE-IDs must be unique per edge router. Look at many options which are already either disabled or enabled. MAC learning and withdraw messaging are enabled which is a recommended configuration for the most of cases. In the Data Center increase the MAC aging time from 300 s to at least 1800 s or even 14400 s (depending on the number of MAC addresses) to avoid excessive flooding.

```
RP/0/0/CPU0:SP1R2#sh l2vpn bridge-domain bd-name bd1 det
Legend: pp = Partially Programmed.
Bridge group: bg1, bridge-domain: bd1, id: 0, state: up, ShgId: 0, MSTi: 0
  Coupled state: disabled
  VINE state: Default
  MAC learning: enabled
  MAC withdraw: enabled
    MAC withdraw for Access PW: enabled
    MAC withdraw sent on: bridge port up
    MAC withdraw relaying (access to access): disabled
  Flooding:
    Broadcast & Multicast: enabled
    Unknown unicast: enabled
  MAC aging time: 300 s, Type: inactivity
  MAC limit: 4000, Action: none, Notification: syslog
  MAC limit reached: no
  MAC port down flush: enabled
  MAC Secure: disabled, Logging: disabled
  Split Horizon Group: none
  Dynamic ARP Inspection: disabled, Logging: disabled
  IP Source Guard: disabled, Logging: disabled
  DHCPv4 snooping: disabled
  IGMP Snooping: enabled
  IGMP Snooping profile: none
  MLD Snooping profile: none
  Storm Control: disabled
  Bridge MTU: 1500
  MIB cvplsConfigIndex: 1
  Filter MAC addresses:
  P2MP PW: disabled
  Create time: 29/04/2017 09:26:53 (00:13:32 ago)
  No status change since creation
  ACs: 1 (1 up), VFIs: 1, PWs: 3 (3 up), PBBs: 0 (0 up), VNIs: 0 (0 up)
  List of ACs:
    AC: GigabitEthernet0/0/0/1.40, state is up
      Type VLAN; Num Ranges: 1
      VLAN ranges: [40, 40]
      MTU 1504; XC ID 0x1; interworking none
      MAC learning: enabled
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Flooding:
        Broadcast & Multicast: enabled
        Unknown unicast: enabled
      MAC aging time: 300 s, Type: inactivity
      MAC limit: 4000, Action: none, Notification: syslog
      MAC limit reached: no
      MAC port down flush: enabled
      MAC Secure: disabled, Logging: disabled
      Split Horizon Group: none
      Dynamic ARP Inspection: disabled, Logging: disabled
      IP Source Guard: disabled, Logging: disabled
      DHCPv4 snooping: disabled
      IGMP Snooping: enabled
      IGMP Snooping profile: none
      MLD Snooping profile: none
      Storm Control: bridge-domain policer
      Static MAC addresses:
      Statistics:
        packets: received 0 (multicast 0, broadcast 0, unknown unicast 0, unicast 0),
sent 0
        bytes: received 0 (multicast 0, broadcast 0, unknown unicast 0, unicast 0), sent
0
        MAC move: 0
      Storm control drop counters:
        packets: broadcast 0, multicast 0, unknown unicast 0
        bytes: broadcast 0, multicast 0, unknown unicast 0
      Dynamic ARP inspection drop counters:
        packets: 0, bytes: 0
      IP source guard drop counters:
        packets: 0, bytes: 0
  List of Access PWs:
  List of VFIs:
    VFI vfl (up)
      VPN-ID: 100, Auto Discovery: BGP, state is  Provisioned (Service Connected
      Route Distinguisher:  4:4
      Import Route Targets:
        40:40
      Export Route Targets:
        40:40
      Signaling protocol: BGP
      Local VE-ID: 2 ,  Advertised Local VE-ID : 2
      VE-Range: 10
      PW: neighbor 33.56.1.1, PW ID 100, state is up ( established )
        PW class not set, XC ID 0xfff80001
        Encapsulation MPLS, Auto-discovered (BGP), protocol BGP
        Source address 33.56.2.2
        PW type VPLS, control word disabled, interworking none
        Sequencing not set

          MPLS         Local                          Remote
          ------------ ------------------------------ ------------------------
          Label        102030                         101016
          MTU          1500                           1500
          Control word disabled                       disabled
          PW type      VPLS                           VPLS
          VE-ID        2                              1
          ------------ ------------------------------ ------------------------
        MIB cpwVcIndex: 4294443009
        Create time: 29/04/2017 09:29:01 (00:11:25 ago)
        Last time status changed: 29/04/2017 09:29:01 (00:11:25 ago)
        MAC withdraw messages: sent 0, received 0
        Forward-class: 0
        Static MAC addresses:
        Statistics:
          packets: received 0 (unicast 0), sent 0
          bytes: received 0 (unicast 0), sent 0
          MAC move: 0
        Storm control drop counters:
          packets: broadcast 0, multicast 0, unknown unicast 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      bytes: broadcast 0, multicast 0, unknown unicast 0
  DHCPv4 snooping: disabled
  IGMP Snooping profile: none
  MLD Snooping profile: none
  PW: neighbor 33.56.3.3, PW ID 100, state is up ( established )
    PW class not set, XC ID 0xfff80002
    Encapsulation MPLS, Auto-discovered (BGP), protocol BGP
    Source address 33.56.2.2
    PW type VPLS, control word disabled, interworking none
    Sequencing not set

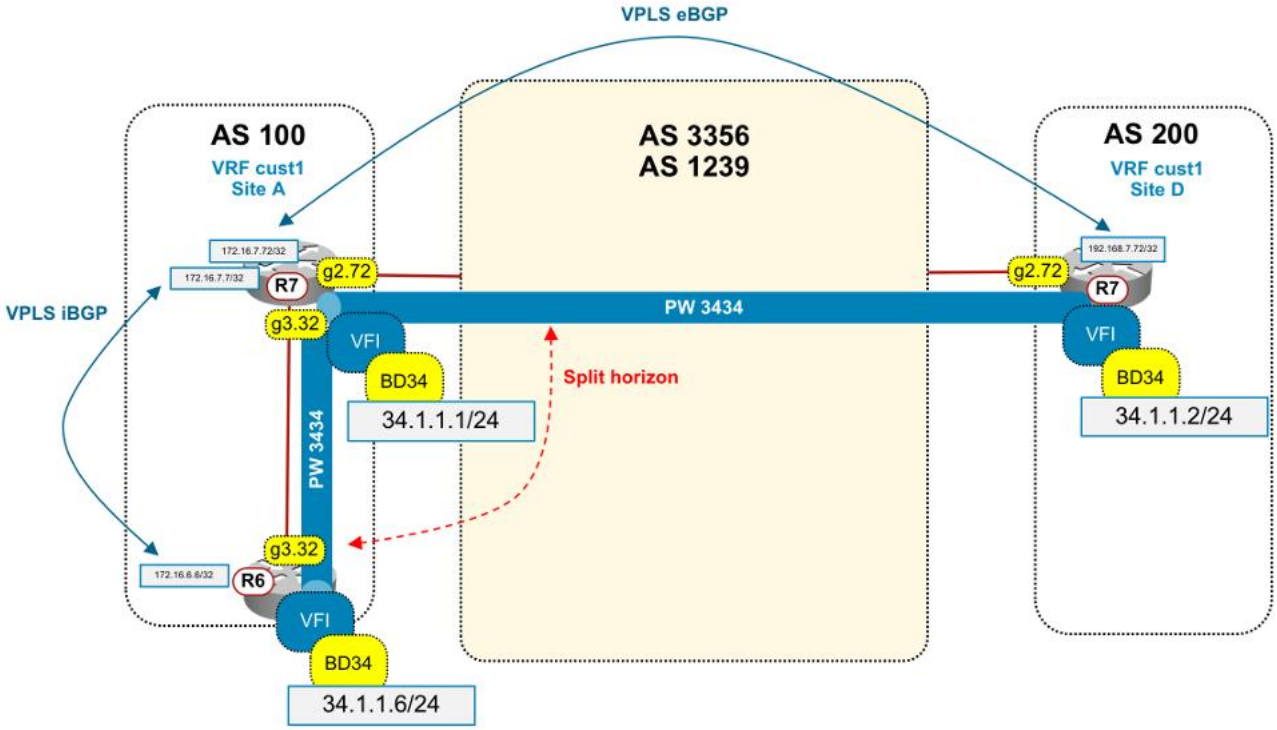
      MPLS         Local                          Remote
      ------------ ------------------------------ ------------------------
      Label        102032                         103031
      MTU          1500                           1500
      Control word disabled                       disabled
      PW type      VPLS                           VPLS
      VE-ID        2                              3
      ------------ ------------------------------ ------------------------
    MIB cpwVcIndex: 4294443010
    Create time: 29/04/2017 09:29:11 (00:11:15 ago)
    Last time status changed: 29/04/2017 09:29:11 (00:11:15 ago)
    MAC withdraw messages: sent 0, received 0
    Forward-class: 0
    Static MAC addresses:
    Statistics:
      packets: received 0 (unicast 0), sent 0
      bytes: received 0 (unicast 0), sent 0
      MAC move: 0
    Storm control drop counters:
      packets: broadcast 0, multicast 0, unknown unicast 0
      bytes: broadcast 0, multicast 0, unknown unicast 0
  DHCPv4 snooping: disabled
  IGMP Snooping profile: none
  MLD Snooping profile: none
  PW: neighbor 33.56.4.4, PW ID 100, state is up ( established )
    PW class not set, XC ID 0xfff80003
    Encapsulation MPLS, Auto-discovered (BGP), protocol BGP
    Source address 33.56.2.2
    PW type VPLS, control word disabled, interworking none
    Sequencing not set

      MPLS         Local                          Remote
      ------------ ------------------------------ ------------------------
      Label        102033                         104076
      MTU          1500                           1500
      Control word disabled                       disabled
      PW type      VPLS                           VPLS
      VE-ID        2                              4
      ------------ ------------------------------ ------------------------
    MIB cpwVcIndex: 4294443011
    Create time: 29/04/2017 09:29:21 (00:11:05 ago)
    Last time status changed: 29/04/2017 09:29:21 (00:11:05 ago)
    MAC withdraw messages: sent 0, received 0
    Forward-class: 0
    Static MAC addresses:
    Statistics:
      packets: received 0 (unicast 0), sent 0
      bytes: received 0 (unicast 0), sent 0
      MAC move: 0
    Storm control drop counters:
      packets: broadcast 0, multicast 0, unknown unicast 0
      bytes: broadcast 0, multicast 0, unknown unicast 0
  DHCPv4 snooping: disabled
  IGMP Snooping profile: none
  MLD Snooping profile: none
  VFI Statistics:
    drops: illegal VLAN 0, illegal length 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: E-TREE BGP signalled VPLS

Build an E-TREE service where spokes R6 (AS 100) and R7 (AS 200) can talk with the hub R7 (AS 100) only. This is a default behaviour of VPLS where the split horizon rule stops the spoke to spoke traffic.



Configure a new VPLS instance signalled by BGP.

```
CEL7(config)#l2vpn vfi context vpls-bgp
CEL7(config-vfi)# vpn id 3434
CEL7(config-vfi)# autodiscovery bgp signaling bgp
CEL7(config-vfi-autodiscovery)#  ve id 1
CEL7(config-vfi-autodiscovery)#  rd 7:7
CEL7(config-vfi-autodiscovery)#  route-target export 3:3
CEL7(config-vfi-autodiscovery)#  bridge-domain 34
CEL7(config-bdomain)#l2vpn vfi context vpls-bgp
CEL7(config-vfi)#  bridge-domain 34
CEL7(config-bdomain)#member vfi vpls-bgp
CEL7(config-bdomain)#int bdi 34
CEL7(config-if)#ip add 34.1.1.1 255.255.255.0
CEL7(config-if)#no shut

CER7(config)#l2vpn vfi context vpls-bgp
CER7(config-vfi)# vpn id 3434
CER7(config-vfi)# autodiscovery bgp signaling bgp
CER7(config-vfi-autodiscovery)#  ve id 2
CER7(config-vfi-autodiscovery)#  rd 7:7
CER7(config-vfi-autodiscovery)#  route-target export 3:3
CER7(config-vfi-autodiscovery)#bridge-domain 34
CER7(config-bdomain)#member vfi vpls-bgp
CER7(config-bdomain)#int bdi 34
CER7(config-if)#ip add 34.1.1.2 255.255.255.0
CER7(config-if)#no shut

CEL6(config)#l2vpn vfi context vpls-bgp
CEL6(config-vfi)# vpn id 3434
CEL6(config-vfi)# autodiscovery bgp signaling bgp
CEL6(config-vfi-autodiscovery)#  ve id 3
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CEL6(config-vfi-autodiscovery)#  ve range 11
CEL6(config-vfi-autodiscovery)#  rd 7:7
CEL6(config-vfi-autodiscovery)#  route-target export 3:3
CEL6(config-vfi-autodiscovery)#  route-target import 3:3
CEL6(config-vfi-autodiscovery)#bridge-domain 34
CEL6(config-bdomain)#member vfi vpls-bgp
CEL6(config-bdomain)#int bdi 34
CEL6(config-if)#ip add 34.1.1.6 255.255.255.0
CEL6(config-if)#no shut
```

Add the VPLS address family in BGP and suppress the LDP signaling as we want to receive VC labels from BGP.

```
CEL7(config)#router bgp 100
CEL7(config-router)#add l2vpn vpls
CEL7(config-router-af)#neighbor 192.168.7.72 activate
CEL7(config-router-af)#neighbor 192.168.7.72 send-community extended
CEL7(config-router-af)#neighbor 192.168.7.72 suppress-signaling-protocol ldp

CER7(config)#router bgp 200
CER7(config-router)#add l2vpn vpls
CER7(config-router-af)#neighbor 172.16.7.72 activate
CER7(config-router-af)#neighbor 172.16.7.72 send-community extended
CER7(config-router-af)#neighbor 172.16.7.72 suppress-signaling-protocol ldp

CEL6(config)#router bgp 100
CEL6(config-router)#add l2vpn vpls
CEL6(config-router-af)#neighbor 172.16.7.7 activate
CEL6(config-router-af)#neighbor 172.16.7.7 send-community extended
CEL6(config-router-af)#neighbor 172.16.7.7 suppress-signaling-protocol ldp
```

BGP verification.

```
CEL7#sh ip bgp l2vpn vpls all summ
BGP router identifier 172.16.7.7, local AS number 100
BGP table version is 10, main routing table version 10
5 network entries using 1320 bytes of memory
5 path entries using 720 bytes of memory
5/5 BGP path/bestpath attribute entries using 1360 bytes of memory
4 BGP AS-PATH entries using 128 bytes of memory
1 BGP community entries using 24 bytes of memory
6 BGP extended community entries using 208 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3760 total bytes of memory
BGP activity 104/88 prefixes, 48809/48793 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.6.6      4          100    1844    1907       10    0    0 1d03h           2
192.168.7.72    4          200    1843    1847       10    0    0 1d03h           1

CEL7#sh ip bgp l2vpn vpls all
BGP table version is 10, local router ID is 172.16.7.7
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 7:7
 *>   7:7:VEID-1:Blk-1/136
                     0.0.0.0                            32768 ?
 *>   7:7:VEID-2:Blk-1/136
                     192.168.7.72             0             0 200 ?
 *>i  7:7:VEID-3:Blk-1/136
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
                     172.16.6.6               0    100      0 ?
Route Distinguisher: 200:1
 *>   200:1:VEID-1:Blk-1/136
                     0.0.0.0                            32768 ?
 *>i  200:1:VEID-2:Blk-1/136
                     172.16.6.6               0    100      0 ?
```

Virtual circuits verification.

```
CEL7#sh mpls l2transport vc

Local intf     Local circuit               Dest address    VC ID      Status
-------------  --------------------------- --------------- ---------- ----------
VFI vpls-bgp   vfi                         192.168.7.72    3434       UP
VFI vpls-bgp   vfi                         172.16.6.6      3434       UP
pw100015       192.168.7.72 3232           172.16.6.6      3232       UP
VFI hvpls      vfi                         172.16.6.6      3367       UP
pw100003       192.168.7.72 60             172.16.6.6      6060       UP
pw100002       172.16.6.6 6060             192.168.7.72    60         UP
Gi3.30         Eth VLAN 30                 192.168.7.72    3030       UP
Gi3.31         Eth VLAN 100/31             192.168.7.72    3131       UP
pw100016       172.16.6.6 3232             192.168.7.72    3232       UP
VFI hvpls      vfi                         192.168.7.72    3367       UP
Gi3.60         Eth VLAN 60                 192.168.7.72    6060       UP


CEL7#sh mpls l2transport vc 3434 det
Local interface: VFI vpls-bgp vfi up
  Interworking type is Ethernet
  Destination address: 192.168.7.72, VC ID: 3434, VC status: up
  Next hop PE address: 192.168.7.72
    Output interface: Gi2.72, imposed label stack {102007 136}
    Preferred path: not configured
    Default path: active
    Next hop: 10.1.27.2
  Create time: 1d03h, last status change time: 1d03h
    Last label FSM state change time: 1d03h
  Signaling protocol: BGP
    Status TLV support (local/remote)   : Not Applicable
      LDP route watch                   : Not Applicable
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not Applicable
      Last BFD peer monitor  status rcvd: Not Applicable
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: Not Applicable
      Last remote LDP TLV    status rcvd: Not Applicable
      Last remote LDP ADJ    status rcvd: Not Applicable
    MPLS VC labels: local 128, remote 136
    Group ID: local n/a, remote n/a
    MTU: local 1500, remote 1500
  Control Word: Off
  SSO Descriptor: 192.168.7.72/3434, local label: 128
  Dataplane:
    SSM segment/switch IDs: 41068/20586 (used), PWID: 16
  VC statistics:
    transit packet totals: receive 0, send 26
    transit byte totals:   receive 0, send 2236
    transit packet drops:  receive 0, seq error 0, send 0

Local interface: VFI vpls-bgp vfi up
  Interworking type is Ethernet
  Destination address: 172.16.6.6, VC ID: 3434, VC status: up
  Next hop PE address: 172.16.6.6
    Output interface: Gi3.32, imposed label stack {75}
    Preferred path: not configured
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    Default path: active
    Next hop: 10.1.67.6
  Create time: 1d03h, last status change time: 1d03h
    Last label FSM state change time: 1d03h
  Signaling protocol: BGP
    Status TLV support (local/remote)   : Not Applicable
      LDP route watch                   : Not Applicable
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not Applicable
      Last BFD peer monitor  status rcvd: Not Applicable
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: Not Applicable
      Last remote LDP TLV    status rcvd: Not Applicable
      Last remote LDP ADJ    status rcvd: Not Applicable
    MPLS VC labels: local 129, remote 75
    Group ID: local n/a, remote n/a
    MTU: local 1500, remote 1500
  Control Word: Off
  SSO Descriptor: 172.16.6.6/3434, local label: 129
  Dataplane:
    SSM segment/switch IDs: 49263/24685 (used), PWID: 17
  VC statistics:
    transit packet totals: receive 0, send 19
    transit byte totals:   receive 0, send 1558
    transit packet drops:  receive 0, seq error 0, send 0
```

You can compare received labels with this output.

```
CEL7#sh ip bgp l2vpn vpls all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 7:7
   7:7:VEID-1:Blk-1/136
                    0.0.0.0         138/127
   7:7:VEID-2:Blk-1/136
                    192.168.7.72    exp-null/136
   7:7:VEID-3:Blk-1/136
                    172.16.6.6      149/75
```

This is how you can check BGP attributes for the VPLS route.

```
CEL7#sh ip bgp l2vpn vpls all ve-id 1 block-offset 1
BGP routing table entry for 7:7:VEID-1:Blk-1/136, version 14
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
  Advertised to update-groups:
     5
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (172.16.7.7)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      AGI version(0), VE Block Size(11) Label Base(87)
      Local Label Base(98) Block ID (6)
      Extended Community: RT:100:3434 RT:3:3 L2VPN L2:0x0:MTU-1500
      mpls labels in/out 98/87
      rx pathid: 0, tx pathid: 0x0
```

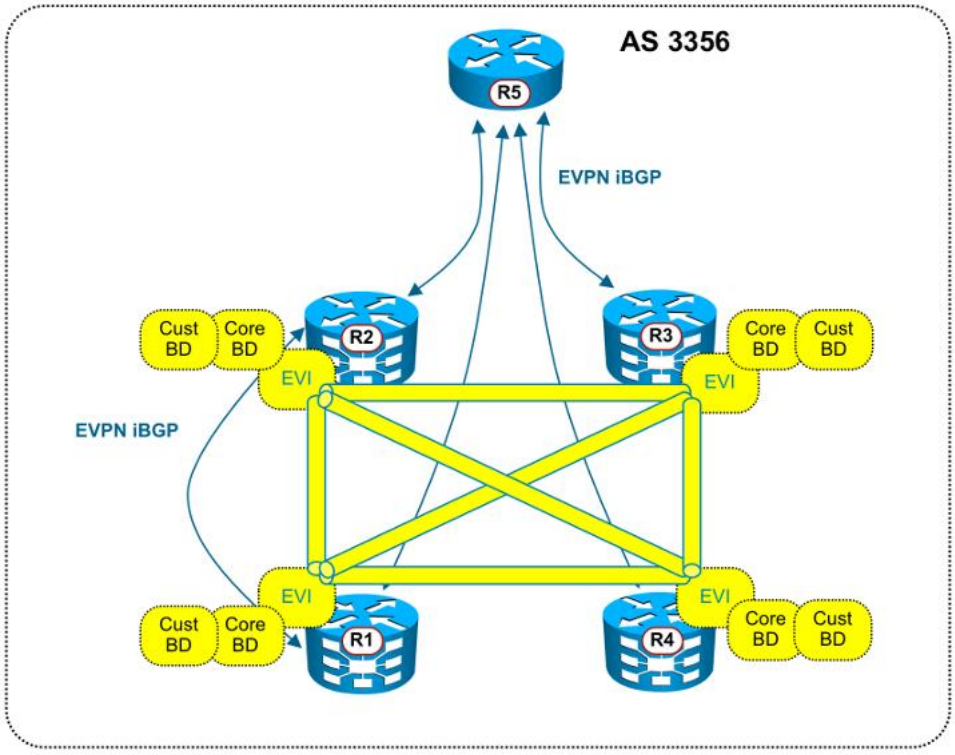
Please note that the data plane for this configuration on virtualized image of IOS XE does not work. You can verify the control-plane.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: PBB-EVPN

Task 2.1: E-LAN BGP signalled PBB-EVPN

Configure an E-LAN service according to the PBB-EVPN topology depicted as follows.



PBB-EVPN has two levels of hierarchy. The first level is MAC-in-MAC which is Provider Backbone Bridges (PBB). The customer Ethernet frame is encapsulated into a special PBB tagging and an external Ethernet header. The second level is VPLS with MPLS encapsulation in our case.

Enable the EVPN AFI in BGP on RR.

```
SP1-R5-RR(config)#router bgp 3356
SP1-R5-RR(config-router)#address-family l2vpn evpn
SP1-R5-RR(config-router-af)#  neighbor 33.56.1.1 activate
SP1-R5-RR(config-router-af)#  neighbor 33.56.1.1 send-community extended
SP1-R5-RR(config-router-af)#  neighbor 33.56.1.1 route-reflector-client
SP1-R5-RR(config-router-af)#  neighbor 33.56.2.2 activate
SP1-R5-RR(config-router-af)#  neighbor 33.56.2.2 send-community extended
SP1-R5-RR(config-router-af)#  neighbor 33.56.2.2 route-reflector-client
SP1-R5-RR(config-router-af)#  neighbor 33.56.3.3 activate
SP1-R5-RR(config-router-af)#  neighbor 33.56.3.3 send-community extended
SP1-R5-RR(config-router-af)#  neighbor 33.56.3.3 route-reflector-client
SP1-R5-RR(config-router-af)#  neighbor 33.56.4.4 activate
SP1-R5-RR(config-router-af)#  neighbor 33.56.4.4 send-community extended
SP1-R5-RR(config-router-af)#  neighbor 33.56.4.4 route-reflector-client
```

Configure all RR clients. Each router has a unique backbone source MAC. This MAC is used as a source MAC in the external Ethernet header when the frame is leaves the backbone bridge and enters the VFI instance. This address is

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

derived from the system of a device automatically. In this task this is set manually.

R1's config example. Set the unique backbone-source-mac.

```
interface GigabitEthernet0/0/0/0.41 l2transport
 encapsulation dot1q 41
 rewrite ingress tag pop 1 symmetric
!
router bgp 3356
 address-family l2vpn evpn
 !
 neighbor 33.56.5.5
  address-family l2vpn evpn
   next-hop-self
  !
 !
!
l2vpn
 pbb
  backbone-source-mac 0001.0001.0001
 !
 bridge group bdc
  bridge-domain bdc
   pbb core
    evi 100
    !
   !
  !
 !
 bridge group bde
  bridge-domain bde
   interface GigabitEthernet0/0/0/0.41
   !
   pbb edge i-sid 4141 core-bridge bdc
   !
  !
 !
!
```

EVPN session verification.

```
SP1-R5-RR#sh bgp l2vpn evpn
BGP table version is 9, local router ID is 33.56.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 33.56.1.1:100
 *>i  [3][33.56.1.1:100][4141][32][33.56.1.1]/17
                      33.56.1.1                100      0 i
Route Distinguisher: 33.56.2.2:100
 *>i  [3][33.56.2.2:100][4141][32][33.56.2.2]/17
                      33.56.2.2                100      0 i
Route Distinguisher: 33.56.3.3:100
 *>i  [3][33.56.3.3:100][4141][32][33.56.3.3]/17
                      33.56.3.3                100      0 i
Route Distinguisher: 33.56.4.4:100
 *>i  [3][33.56.4.4:100][4141][32][33.56.4.4]/17
                      33.56.4.4                100      0 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

As an additional subtask configure a direct EVPN session between R1 and R2 so you have the following result with route type-2 learnt from R1.

```
RP/0/0/CPU0:SP1R2#sh bgp l2vpn evpn
[..]
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 33.56.1.1:100
*>i[2][0][48][0001.0001.0001][0]/104
                      33.56.1.1                100      0 i
*>i[3][4141][32][33.56.1.1]/80
                      33.56.1.1                100      0 i
* i                   33.56.1.1                100      0 i
Route Distinguisher: 33.56.2.2:100 (default for vrf bdc)
*>i[2][0][48][0001.0001.0001][0]/104
                      33.56.1.1                100      0 i
*> [2][0][48][0002.0002.0002][0]/104
                      0.0.0.0                           0 i
*>i[3][4141][32][33.56.1.1]/80
                      33.56.1.1                100      0 i
*> [3][4141][32][33.56.2.2]/80
                      0.0.0.0                           0 i
*>i[3][4141][32][33.56.3.3]/80
                      33.56.3.3                100      0 i
*>i[3][4141][32][33.56.4.4]/80
                      33.56.4.4                100      0 i
Route Distinguisher: 33.56.3.3:100
*>i[3][4141][32][33.56.3.3]/80
                      33.56.3.3                100      0 i
Route Distinguisher: 33.56.4.4:100
*>i[3][4141][32][33.56.4.4]/80
                      33.56.4.4                100      0 i

Processed 10 prefixes, 11 paths
```

Verify that the backbone bridge of R2 learnt the backbone bridge's MAC address of R1.

```
RP/0/0/CPU0:SP1R2#sh l2vpn forwarding bridge-domain bdc:bdc mac-address det location
0/0/CPU0
[..]
Bridge-domain name: bdc:bdc, id: 1, state: up
 Type: pbb-core
 Number of associated pbb-edge BDs: 1
 MAC learning: enabled
 MAC port down flush: enabled
 Flooding:
   Broadcast & Multicast: enabled
   Unknown unicast: enabled
[..]

    Mac Address: 0001.0001.0001, LC learned: N/A
       Resync Age: N/A, Flag: static, BMAC
       Next hop: 33.56.1.1, Label: 101005

  PBB Core, state: Up
    XC ID: 0xfff00001
    Number of MAC: 0 (SW), 0 (HW)

  EVPN, state: Up
    Evi: 100
    XC ID: 0xfff01001
    Number of MAC: 0 (SW), 0 (HW)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

MPLS L2VPN: all services listed together (verification):

```
CEL7#sh mpls l2 vc

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
VFI vpls-bgp   vfi                        192.168.7.72    3434       UP
VFI vpls-bgp   vfi                        172.16.6.6      3434       UP
pw100015       192.168.7.72 3232          172.16.6.6      3232       UP
VFI hvpls      vfi                        172.16.6.6      3367       UP
pw100003       192.168.7.72 60            172.16.6.6      6060       UP
pw100002       172.16.6.6 6060            192.168.7.72    60         UP
Gi3.30         Eth VLAN 30                192.168.7.72    3030       UP
Gi3.31         Eth VLAN 100/31            192.168.7.72    3131       UP
pw100016       172.16.6.6 3232            192.168.7.72    3232       UP
VFI hvpls      vfi                        192.168.7.72    3367       UP
Gi3.60         Eth VLAN 60                192.168.7.72    6060       UP

CEL7#sh l2vpn service all
Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down              AD=Admin Down   IA=Inactive
        SB=Standby  HS=Hot Standby       RV=Recovering   NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                  Prio  St  XC St
  ---------          -----       -------------                  ----  --  -----
VPWS name: Gi3.30-30, State: UP
  Gi3.30             left        Gi3.30:30(Eth VLAN)            0     UP  UP
  pw100017           right       192.168.7.72:3030(MPLS)        0     UP  UP
VPWS name: Gi3.31-100/31, State: UP
  Gi3.31             left        Gi3.31:100/31(Eth VLAN)        0     UP  UP
  pw100018           right       192.168.7.72:3131(MPLS)        0     UP  UP
VPWS name: Gi3.37-37, State: UP
  Gi3.37             left        Gi3.37:37(Eth VLAN)            0     UP  UP
  pw100007           right       192.168.7.72:3737(L2TP)        0     UP  UP
VPWS name: Gi3.60-60, State: UP
  Gi3.60             left        Gi3.60:60(Eth VLAN)            0     UP  UP
  pw100001           right       192.168.7.72:6060(MPLS)        0     UP  UP
VPWS name: mpw, State: UP
  pw100015                       192.168.7.72:3232(MPLS)        0     UP  UP
  pw100016                       172.16.6.6:3232(MPLS)          0     UP  UP
VPWS name: p1, State: UP
  pw100002                       172.16.6.6:6060(MPLS)          0     UP  UP
  pw100003                       192.168.7.72:60(MPLS)          0     UP  UP
VPLS name: hvpls, State: UP
  pw100012                       hvpls(VFI)                     0     UP  UP
  pw100014           core_pw     192.168.7.72:3367(MPLS)        0     UP  UP
  pw100013           core_pw     172.16.6.6:3367(MPLS)          0     UP  UP
VPLS name: vpls-bgp, State: UP
  pw100009                       vpls-bgp(VFI)                  0     UP  UP
  pw100011           core_pw     3:3434(MPLS)                   0     UP  UP
  pw100010           core_pw     2:3434(MPLS)                   0     UP  UP
BD name: 33, State: --
  -                              33(BD)                         0     UP  --
  -                              hvpls(VFI)                     0     UP  UP
BD name: 34, State: --
  -                              34(BD)                         0     UP  --
  -                              vpls-bgp(VFI)                  0     UP  UP


CER7#sh mpls l2transport vc

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
VFI vpls-bgp   vfi                        172.16.7.72     3434       UP
VFI vpls-bgp   vfi                        172.16.7.72     3434       UP
pw100010       192.168.6.6 6060           172.16.7.72     60         UP
Gi3.30         Eth VLAN 30                172.16.7.72     3030       UP
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Gi3.31          Eth VLAN 100/31             172.16.7.72     3131        UP
pw100005        192.168.6.6 3232            172.16.7.72     3232        UP
VFI hvpls       vfi                         172.16.7.72     3367        UP
Gi3.60          Eth VLAN 60                 172.16.7.72     6060        UP
pw100004        172.16.7.72 3232            192.168.6.6     3232        UP
pw100011        172.16.7.72 60              192.168.6.6     6060        UP

CER7#sh l2vpn service all
Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down             AD=Admin Down    IA=Inactive
        SB=Standby  HS=Hot Standby      RV=Recovering    NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                   Prio  St  XC St
  ---------          -----       -------------                   ----  --  -----
VPWS name: Gi3.30-30, State: UP
  Gi3.30             left        Gi3.30:30(Eth VLAN)             0     UP  UP
  pw100006           right       172.16.7.72:3030(MPLS)          0     UP  UP
VPWS name: Gi3.31-100/31, State: UP
  Gi3.31             left        Gi3.31:100/31(Eth VLAN)         0     UP  UP
  pw100007           right       172.16.7.72:3131(MPLS)          0     UP  UP
VPWS name: Gi3.37-37, State: UP
  Gi3.37             left        Gi3.37:37(Eth VLAN)             0     UP  UP
  pw100012           right       172.16.7.72:3737(L2TP)          0     UP  UP
VPWS name: Gi3.60-60, State: UP
  Gi3.60             left        Gi3.60:60(Eth VLAN)             0     UP  UP
  pw100009           right       172.16.7.72:6060(MPLS)          0     UP  UP
VPWS name: mpw, State: UP
  pw100004                       172.16.7.72:3232(MPLS)          0     UP  UP
  pw100005                       192.168.6.6:3232(MPLS)          0     UP  UP
VPWS name: p1, State: UP
  pw100010                       192.168.6.6:6060(MPLS)          0     UP  UP
  pw100011                       172.16.7.72:60(MPLS)            0     UP  UP
VPLS name: hvpls, State: UP
  pw100002                       hvpls(VFI)                      0     UP  UP
  pw100003           core_pw     172.16.7.72:3367(MPLS)          0     UP  UP
VPLS name: vpls-bgp, State: UP
  pw100001                       vpls-bgp(VFI)                   0     UP  UP
  pw100014           core_pw     3:3434(MPLS)                    0     UP  UP
  pw100013           core_pw     1:3434(MPLS)                    0     UP  UP
BD name: 33, State: --
  -                              33(BD)                          0     UP  --
  -                              hvpls(VFI)                      0     UP  UP
BD name: 34, State: --
  -                              34(BD)                          0     UP  --
  -                              vpls-bgp(VFI)                   0     UP  UP


CEL6#sh l2vpn service all
Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down             AD=Admin Down    IA=Inactive
        SB=Standby  HS=Hot Standby      RV=Recovering    NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                   Prio  St  XC St
  ---------          -----       -------------                   ----  --  -----
VPWS name: Gi2-SI:132:132, State: UP
  Gi2                left        Gi2:102(Eth VLAN)               0     UP  UP
  pw100002           right       172.16.7.7:3232(MPLS)           0     UP  UP
VPWS name: Gi2-SI:33:33, State: UP
  Gi2                left        Gi2:33(Eth VLAN)                0     UP  UP
  pw100005           right       172.16.7.7:3367(MPLS)           0     UP  UP
VPWS name: Gi2-SI:60:61, State: UP
  Gi2                left        Gi2:61(Eth VLAN)                0     UP  UP
  pw100009           right       172.16.7.7:6060(MPLS)           0     UP  UP
VPLS name: vpls-bgp, State: UP
  pw100013                       vpls-bgp(VFI)                   0     UP  UP
  pw100015           core_pw     2:3434(MPLS)                    0     DN  DN
  pw100014           core_pw     1:3434(MPLS)                    0     UP  UP
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
BD name: 34, State: --
  -                             34(BD)                          0     UP  --
  -                             vpls-bgp(VFI)                   0     UP  UP

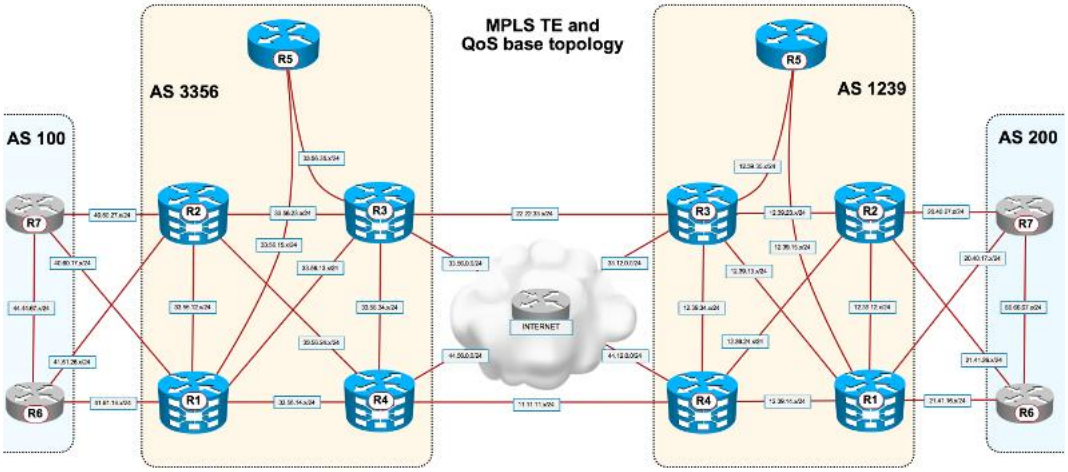

CER6#sh l2vpn service all
Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down             AD=Admin Down     IA=Inactive
        SB=Standby  HS=Hot Standby      RV=Recovering     NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                   Prio  St  XC St
  ---------          -----       -------------                   ----  --  -----
VPWS name: Gi2-SI:132:132, State: UP
  Gi2                left        Gi2:101(Eth VLAN)               0     UP  UP
  pw100001           right       192.168.7.7:3232(MPLS)          0     UP  UP
VPWS name: Gi2-SI:60:60, State: UP
  Gi2                left        Gi2:60(Eth VLAN)                0     UP  UP
  pw100003           right       192.168.7.7:6060(MPLS)          0     UP  UP
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 12: MPLS Traffic Engineering

For MPLS Traffic Engineering we'll use base topology you've build during Session 4, but with the configs from finished Session 6 (that include MPLS L3VPN and L2VPN). Basic topology is therefore pictured below:



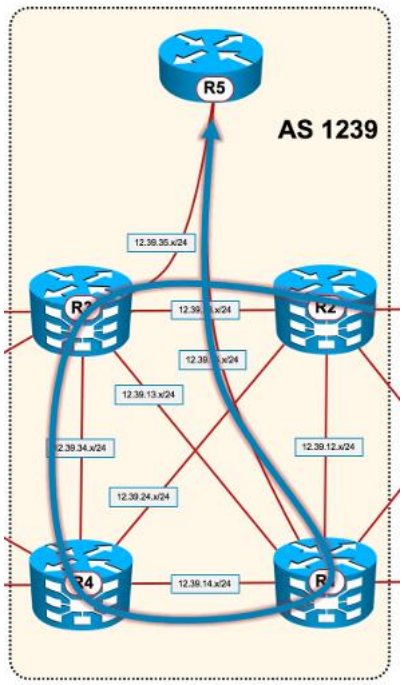
This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch12.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch12.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch12.unl`

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: Basic MPLS TE configuration

We'll setup simple MPLS TE tunnel within AS1239. The idea is to force traffic going to SP2R5 from SP2R2 via set of routers within AS1239:



Traffic should follow following explicit path: R2->R3->R4->R1->R5. By default, traffic to 12.39.5.5 from R2 goes via IGP load balanced path of:

```
RP/0/0/CPU0:SP2R2#show ip route 12.39.5.5
Routing entry for 12.39.5.5/32
  Known via "isis AS1239", distance 115, metric 30, type level-2
  Installed Jan 01 23:39:41.301 for 16:16:22
  Routing Descriptor Blocks
    12.39.23.3, from 12.39.5.5, via GigabitEthernet0/0/0/0
      Route metric is 30
    12.39.12.1, from 12.39.5.5, via GigabitEthernet0/0/0/2
      Route metric is 30
  No advertising protos.
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.1: Enabling RSVP on interfaces

By default, if you don't want to specify RSVP parameters, you can enable RSVP by enabling MPLS TE on interfaces like that:

```
RP/0/0/CPU0:SP2R2#show rsvp int
RSVP process is not running

RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#mpls traffic-eng
RP/0/0/CPU0:SP2R2(config-mpls-te)#int gi0/0/0/0
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#int gi0/0/0/2
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#int gi0/0/0/3
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#commit
RP/0/0/CPU0:SP2R2#show rsvp int

*: RDM: Default I/F B/W % : 75% [default] (max resv/bc0), 0% [default] (bc1)

Interface                MaxBW (bps)  MaxFlow (bps) Allocated (bps)      MaxSub (bps)
------------------------ ------------ ------------- -------------------- -------------
GigabitEthernet0/0/0/0              0             0             0 (  0%)             0
GigabitEthernet0/0/0/2              0             0             0 (  0%)             0
GigabitEthernet0/0/0/3              0             0             0 (  0%)             0
RP/0/0/CPU0:SP2R2#
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: Interface parameters from RSVP-TE point of view

If the configuration is defined as above, only tunnels with no requested bandwidth will be set up successfully. To change that, let's configure RSVP explicitly and define them as having 10Gbit/s bandwidth.

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#rsvp
RP/0/0/CPU0:SP2R2(config-rsvp)#int gi0/0/0/0
RP/0/0/CPU0:SP2R2(config-rsvp-if)#bandwidth 10 Gbps
RP/0/0/CPU0:SP2R2(config-rsvp-if)#int gi0/0/0/2
RP/0/0/CPU0:SP2R2(config-rsvp-if)#bandwidth 10 Gbps
RP/0/0/CPU0:SP2R2(config-rsvp-if)#int gi0/0/0/3
RP/0/0/CPU0:SP2R2(config-rsvp-if)#bandwidth 10 Gbps
RP/0/0/CPU0:SP2R2(config-rsvp-if)#commit
```

Now, the interface summary from RSVP point of view looks much more like real life:

```
RP/0/0/CPU0:SP2R2#show rsvp int

*: RDM: Default I/F B/W % : 75% [default] (max resv/bc0), 0% [default] (bc1)

Interface                MaxBW (bps)  MaxFlow (bps) Allocated (bps)      MaxSub (bps)
------------------------ ------------ ------------- -------------------- -------------
GigabitEthernet0/0/0/0            10G            10G              0 (   0%)             0
GigabitEthernet0/0/0/2            10G            10G              0 (   0%)             0
GigabitEthernet0/0/0/3            10G            10G              0 (   0%)             0
```

Let's take a look at the default interface parameters from RSVP point of view:

```
RP/0/0/CPU0:SP2R2#show rsvp interface detail

*: RDM: Default I/F B/W % : 75% [default] (max resv/bc0), 0% [default] (bc1)

INTERFACE: GigabitEthernet0/0/0/0 (ifh=0x40).
 VRF ID: 0x60000000 (Default).
 BW (bits/sec): Max=10G. MaxFlow=10G.
                Allocated=0 (0%). MaxSub=1G.
 Signalling: No DSCP marking. No rate limiting.
 States in: 0. Max missed msgs: 4.
 Max out-of-band missed msgs: 38000.
 Expiry timer: Not running. Refresh interval: 45s.
 Normal Refresh timer: Not running. Out-of-band refresh interval: 0s.
 Summary refresh timer: Not running.
 Refresh reduction local: Enabled. Summary Refresh: Enabled (1472 bytes max).
 Reliable summary refresh: Disabled. Bundling: Enabled. (1500 bytes max).
 Ack hold: 400 ms, Ack max size: 1500 bytes. Retransmit: 2100ms.
```

As per configuration, RSVP signals 10Gbit/s of useable bandwidth, with maximum reservation of flow for 10Gbit/s. At the same time, right now from RSVP point of view there's 0% usage of the interface bandwidth. Also, RSVP refreshes state on the interface each 45 seconds.

Repeat the RSVP configuration on SP2R1, SP2R3 and SP2R4.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3: Building dynamic tunnel

Let's setup our first dynamic tunnel from SP2R2 to SP2R5 using all defaults.

Task 1.3.1: Enabling RSVP on all core interfaces

Please remember, to turn on RSVP on all internal interfaces within AS1239 before continuing.

SP2R5 is running IOS-XE, so the configuration needs explicit RSVP definition:

```
SP2-R5-RR(config)#int gi2
SP2-R5-RR(config-if)#ip rsvp bandwidth 10000000
SP2-R5-RR(config-if)#int gi3
SP2-R5-RR(config-if)#ip rsvp bandwidth 10000000
SP2-R5-RR(config-if)#^Z
SP2-R5-RR#show ip rsvp interface
interface    rsvp       allocated  i/f max  flow max sub max  VRF
Gi2          ena        0          10G      10G      0
Gi3          ena        0          10G      10G      0
```

Task 1.3.2: Enabling IGP for MPLS Traffic Engineering

Next step is to let IGP know to signal TE information to RSVP.

On SP2R1, R2, R3 and R4:

```
RP/0/0/CPU0:SP2R1#conf t
RP/0/0/CPU0:SP2R1(config)#router isis AS1239
RP/0/0/CPU0:SP2R1(config-isis)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R1(config-isis-af)#metric-style wide
RP/0/0/CPU0:SP2R1(config-isis-af)#mpls traffic-eng level 2
RP/0/0/CPU0:SP2R1(config-isis-af)#mpls traffic-eng router-id Loopback0
RP/0/0/CPU0:SP2R1(config-isis-af)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
RP/0/0/CPU0:SP2R1#
```

On SP2R5:

```
SP2-R5-RR#conf t
SP2-R5-RR(config-if)#router isis AS1239
SP2-R5-RR(config-router)#mpls traffic-eng router-id loopback 0
SP2-R5-RR(config-router)#mpls traffic-eng level-2
SP2-R5-RR(config-router)#metric-style wide
SP2-R5-RR(config-router)#^Z
SP2-R5-RR#
```

If the configuration was done correctly, you should have all routers signalling the RSVP capability within the flood domain:

```
RP/0/0/CPU0:SP2R2#show mpls traffic-eng topology isis | i MPLS TE Id
IGP Id: 0000.0000.0001.00, MPLS TE Id: 12.39.1.1 Router Node  (IS-IS AS1239 level-2)
IGP Id: 0000.0000.0002.00, MPLS TE Id: 12.39.2.2 Router Node  (IS-IS AS1239 level-2)
IGP Id: 0000.0000.0003.00, MPLS TE Id: 12.39.3.3 Router Node  (IS-IS AS1239 level-2)
IGP Id: 0000.0000.0004.00, MPLS TE Id: 12.39.4.4 Router Node  (IS-IS AS1239 level-2)
IGP Id: 0000.0000.0005.00, MPLS TE Id: 12.39.5.5 Router Node  (IS-IS AS1239 level-2)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3.3: Enabling MPLS Traffic Engineering on IOS-XE devices

There's one more thing that's needed on SP2R5 (IOS-XE) router to finish the configuration – it's enabling MPLS TE features:

```
SP2-R5-RR(config)#mpls traffic-eng tunnels
SP2-R5-RR(config)#interface gi2
SP2-R5-RR(config-if)#mpls traffic-eng tunnels
SP2-R5-RR(config-if)#interface gi3
SP2-R5-RR(config-if)#mpls traffic-eng tunnels
```



There is subtle but important difference between IOS/IOS-XE and IOS XR devices in terms of signalling labels on second to last LSR (PHP). IOS and IOS-XE treats both labels 0 and 3 as implicit null, and thus the label is not sent - this may be a problem for all signalling using MPLS label.

To force signalling of explicit null for MPLS TE on Cisco IOS and IOS XE configure:

```
mpls traffic-eng signalling interpret explicit-null verbatim
```

Task 1.3.4: Configuring dynamic MPLS TE tunnel

So let's get back to SP2R2 and configure default, dynamic MPLS TE tunnel to SP2R5. We'll request 5Mbps of bandwidth for this tunnel, leaving all other settings default right now:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#interface tunnel-te 25
RP/0/0/CPU0:SP2R2(config-if)#signalled-bandwidth 5000
RP/0/0/CPU0:SP2R2(config-if)#ipv4 unnumbered loo0
RP/0/0/CPU0:SP2R2(config-if)#destination 12.39.5.5
RP/0/0/CPU0:SP2R2(config-if)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

The interface is down:

```
RP/0/0/CPU0:SP2R2#show interfaces tunnel-te 25
tunnel-te25 is down, line protocol is down
  Interface state transitions: 0
  Hardware is Tunnel-TE
  Internet address is 12.39.2.2/32
  MTU 1500 bytes, BW 5000 Kbit (Max: 5000 Kbit)
     reliability 255/255, txload Unknown, rxload Unknown
  Encapsulation TUNNEL,  loopback not set,
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

And that's because we don't have any path option defined. As soon as we'll add path option (for now – dynamic), the tunnel will be signalled up:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#int tunnel-te 25
RP/0/0/CPU0:SP2R2(config-if)#path-option 10 dynamic
```

Let's verify the interface status:

```
RP/0/0/CPU0:SP2R2#show interfaces tunnel-te 25
tunnel-te25 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Tunnel-TE
  Internet address is 12.39.2.2/32
```

And now signalled tunnel status:

```
RP/0/0/CPU0:SP2R2#show mpls traffic-eng tunnels
Name: tunnel-te25  Destination: 12.39.5.5  Ifhandle:0xb0
  Signalled-Name: SP2R2_t25
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type dynamic  (Basis for Setup, path weight 20)
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 5000 kbps  CT0
    Creation Time: Mon Jan  1 16:49:25 2017 (00:04:53 ago)
  Config Parameters:
    Bandwidth:     5000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  History:
    Tunnel has been up for: 00:01:50 (since Mon Jan 01 16:52:28 UTC 2017)
    Current LSP:
      Uptime: 00:01:50 (since Mon Jan 01 16:52:28 UTC 2017)

  Path info (IS-IS AS1239 level-2):
  Node hop count: 2
  Hop0: 12.39.12.1
  Hop1: 12.39.15.5
  Hop2: 12.39.5.5
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

We can see that there's path going from R2 to R1 (12.39.12.1), to R5 interface and then to R5 loopback (our destination).

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

However, the routing table still shows normal forwarding for the destination prefix of 12.39.5.5:

```
RP/0/0/CPU0:SP2R2#show ip ro 12.39.5.5
Routing entry for 12.39.5.5/32
  Known via "isis AS1239", distance 115, metric 30, type level-2
  Installed Mon Jan 01 23:39:41.301 for 17:18:11
  Routing Descriptor Blocks
    12.39.23.3, from 12.39.5.5, via GigabitEthernet0/0/0/0
      Route metric is 30
    12.39.12.1, from 12.39.5.5, via GigabitEthernet0/0/0/2
      Route metric is 30
  No advertising protos.
```

We need to either announce the path via Tunnel or force some specific traffic to Tunnel. Let's announce the tunnel via IGP locally just for the router:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#interface tunnel-te25
RP/0/0/CPU0:SP2R2(config-if)#autoroute announce
RP/0/0/CPU0:SP2R2(config-if-tunte-aa)#commit
```

Now it's in the routing table of this specific router:

```
RP/0/0/CPU0:SP2R2#show ip ro 12.39.5.5
Routing entry for 12.39.5.5/32
  Known via "isis AS1239", distance 115, metric 30, type level-2
  Installed Jan  1 17:01:14.679 for 00:00:26
  Routing Descriptor Blocks
    12.39.5.5, from 12.39.5.5, via tunnel-te25
      Route metric is 30
  No advertising protos.
```

You can check summary brief of all MPLS TE tunnels on node:

```
RP/0/0/CPU0:SP2R2#show mpls traffic-eng tunnels brief

                      TUNNEL NAME          DESTINATION      STATUS  STATE
                      tunnel-te25            12.39.5.5          up  up
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

And to check the tunnels label forwarding information, use:

```
RP/0/0/CPU0:SP2R2#show mpls forwarding tunnels
Tunnel        Outgoing    Outgoing     Next Hop        Bytes
Name          Label       Interface                    Switched
------------- ----------- ------------ --------------- ------------
tt25               123110    Gi0/0/0/2      12.39.12.1             3
```

The other way to look at the RSVP signalled tunnel is to ask RSVP directly:

```
RP/0/0/CPU0:SP2R2#sh rsvp session detail
SESSION: IPv4-LSP Addr: 12.39.5.5, TunID: 25, ExtID: 12.39.2.2
 PSBs: 1, RSBs: 1, Requests: 0
 LSPId: 2
 Tunnel Name: SP2R2_t25
  RSVP Path Info:
   InLabel: No intf, No label
   Incoming Address: Unknown
   Explicit Route:
     Strict, 12.39.12.1/32
     Strict, 12.39.15.5/32
     Strict, 12.39.5.5/32
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
   Record Route: None
   Tspec: avg rate=50M, burst=1K, peak rate=50M
  RSVP Resv Info:
   OutLabel: GigabitEthernet0/0/0/2, 123110
   FRR OutLabel: No intf, No label
   Record Route: None
   Fspec: avg rate=50M, burst=1K, peak rate=50M
```

Let's trace the route from SP2R2 to 12.39.5.5, so tail-end of tunnel:

```
RP/0/0/CPU0:SP2R2#traceroute 12.39.5.5

Type escape sequence to abort.
Tracing the route to 12.39.5.5

 1  12.39.12.1 [MPLS: Label 123110 Exp 0] 0 msec  0 msec  0 msec
 2  12.39.15.5 19 msec  *  0 msec
```

We see that traffic is sent to R1 and then to R5. We also see the label assigned by RSVP – **123110** which matches the label information we've got before. The other option to get more extensive set of information is by doing:

```
RP/0/0/CPU0:SP2R2#show mpls traffic-eng forwarding
P2P tunnels:

Tunnel ID                 Ingress IF     Egress IF      In lbl  Out lbl         Backup
------------------------- -------------- -------------- ------- --------------- --------
12.39.2.2 25_2                       -      Gi0/0/0/2 123221  123110          unknown

Displayed 1 tunnel heads, 0 label P2P rewrites
Displayed 0 tunnel heads, 0 label P2MP rewrites
```

While the '**in label**' will be missing from normal MPLS forwarding table, we can check the label daemon:

```
RP/0/0/CPU0:SP2R2#show mpls lsd forwarding labels 123221
In_Label, (ID), Path_Info: <Type>
123221, (TEv4, 'default':4U, src=12.39.2.2, dst=12.39.5.5, tun_id=25,
ext_tun_id=0x202270c, lsp_id=2), 1 Paths
   1/1: IPv4-Stack (TE), 'default':4U, Gi0/0/0/2, nh=12.39.12.1, lbls={ 123110 },
            flags=0x0, ext_flags=0x40 (TE-Stack)
```

We can also verify with the `tcpdump` that indeed when pinging 12.39.5.5 from SP2R2 we'll label the outgoing traffic with 123110:

```
→  213 20.952223    12.39.12.2      12.39.5.5       ICMP    118 Echo (ping) request  id=0xc0cf, seq=0/0, ttl=255 (reply in 214)
←  214 20.968845    12.39.5.5       12.39.12.2      ICMP    114 Echo (ping) reply    id=0xc0cf, seq=0/0, ttl=254 (request in 213)
▸ Frame 213: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
▸ Ethernet II, Src: fa:16:3e:0d:76:46 (fa:16:3e:0d:76:46), Dst: fa:16:3e:12:98:88 (fa:16:3e:12:98:88)
▾ MultiProtocol Label Switching Header, Label: 123110, Exp: 0, S: 1, TTL: 255
    0001 1110 0000 1110 0110 .... .... .... = MPLS Label: 123110
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1111 = MPLS TTL: 255
▸ Internet Protocol Version 4, Src: 12.39.12.2, Dst: 12.39.5.5
▸ Internet Control Message Protocol
```

In troubleshooting situations, what may be of use is to exploit link state nature of the underlying IGP – whole area within flood domain should be visible to all routers.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

So, from SP2R2 we can check how the TE information looks from SP2R5 point of view:

```
RP/0/0/CPU0:SP2R2#show mpls traffic-eng topology 12.39.5.5 | utility egrep "MPLS|Link"
IGP Id: 0000.0000.0005.00, MPLS TE Id: 12.39.5.5 Router Node  (IS-IS AS1239 level-2)
  Link[0]:Point-to-Point, Nbr IGP Id:0000.0000.0001.00, Nbr Node Id:6, gen:26024
  Link[1]:Point-to-Point, Nbr IGP Id:0000.0000.0003.00, Nbr Node Id:8, gen:26025
```

If you want to check the troubleshooting information about RSVP TE setup (PATH and RESV messages exchange), on IOS XR you'd enable debug for them:

```
RP/0/0/CPU0:SP2R2#debug rsvp path
RP/0/0/CPU0:SP2R2#debug rsvp resv
```

Then the logging information will contain path setup information:

```
RP/0/0/CPU0:SP2R2#show log
[…]
RP/0/0/CPU0:Jan  1 21:22:36.596 : rsvp[1049]: DBG-Path[1]: Creating path (P2P LSP:25,
12.39.2.2 -> 12.39.5.5, 3), fl 0xc0000005, outif 0x40
RP/0/0/CPU0:Jan  1 21:22:36.596 : rsvp[1049]: DBG-Path[1]: Outgoing state: PSB (P2P
LSP:25, 12.39.2.2 -> 12.39.5.5, 3) 0xc0000005, i/f 0x40 i/f Name: Unknown, nh 12.39.23.3
nbr
[…]
RP/0/0/CPU0:Jan  1 21:22:36.596 : rsvp[1049]: DBG-Path[1]: Update PFC ERO next hop:
12.39.23.3
RP/0/0/CPU0:Jan  1 21:22:36.596 : rsvp[1049]: DBG-Path[1]: Path forard for (P2P LSP:25,
12.39.2.2 -> 12.39.5.5, 3); filter: 12.39.2.2
RP/0/0/CPU0:Jan  1 21:22:36.596 : rsvp[1049]: DBG-Path[1]: Forwarding: src 12.39.2.2,
dst 12.39.5.5, nhop 12.39.23.3, i/f GigabitEthernet0/0/0/0
```



If you need to bounce the tunnel to get additional debugging information, you can request re-setup instead of manually shutting it down and then unshutting. On IOS XR you can do this either by name or id:

```
RP/0/0/CPU0:SP1R1#mpls traffic-eng resetup name [NAME-OF-TUNNEL]
RP/0/0/CPU0:SP1R1#mpls traffic-eng resetup p2p tunnel-id ?
  <0-65535>  ID of the P2P tunnel
```

On IOS and IOS-XE you can only request reoptimization by doing:

```
mpls traffic-eng reoptimize tunnel [ID-OF-TUNNEL]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3.5: IGP cost vs TE cost

MPLS TE tunnels are setup by default using TE metrics (within CSPF) but those metrics are reflection of underlying IGP metrics.

The tunnel was setup from SP2R2, to SP2R3 and then to SP2R5 - because we're using uniform cost of 10 for all IGP links. As tunnel spans two links, it's visible as metric 20 here:

```
RP/0/0/CPU0:SP2R2#show mpls traffic-eng tunnels
Name: tunnel-te25  Destination: 12.39.5.5  Ifhandle:0xb0
  Signalled-Name: SP2R2_t25
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type dynamic  (Basis for Setup, path weight 20)
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 5000 kbps  CT0
```

Let's adjust admin weight for TE on Gi0/0/0/2 interface on SP2R2 and switch tunnel to use this metric for path setup. As the path option is dynamic, tunnel should be resetup using any other path avoiding this link:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#mpls traffic-eng
RP/0/0/CPU0:SP2R2(config-mpls-te)#interface gigabitEthernet 0/0/0/2
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#admin-weight 5000
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#commit
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#
```

This should be immediately reflected in adjusted metric for TE:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng topology 12.39.2.2

IGP Id: 0000.0000.0002.00, MPLS TE Id: 12.39.2.2 Router Node  (IS-IS AS1239 level-2)

  Link[0]:Point-to-Point, Nbr IGP Id:0000.0000.0001.00, Nbr Node Id:6, gen:23087
      Frag Id:0, Intf Address:12.39.12.2, Intf Id:0
      Nbr Intf Address:12.39.12.1, Nbr Intf Id:0
      TE Metric:5000, IGP Metric:10
[...]
```

By default, TE metric is choosen and using IGP would require explicit reconfiguration. For our purpose, we'll configure TE metric explicitly on SP2R2 to showcase the way of configuring this:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#interface tunnel-te25
RP/0/0/CPU0:SP2R2(config-if)#path-selection metric te
RP/0/0/CPU0:SP2R2(config-if)#commit
RP/0/0/CPU0:SP2R2(config-if)#
```

Now either wait for tunnel to be reoptimized, or force resetup:

```
RP/0/0/CPU0:SP2R2#mpls traffic-eng resetup name SP2R2_t25
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The tunnel should choose other path:

```
RP/0/0/CPU0:SP2R2#traceroute 12.39.5.5

Type escape sequence to abort.
Tracing the route to 12.39.5.5

 1  12.39.23.3 [MPLS: Label 123307 Exp 0] 0 msec  0 msec  0 msec ! SP2R3
 2  12.39.35.5 0 msec  *  0 msec
RP/0/0/CPU0:SP2R2#
```

And this can be also verified on the tunnel itself:

```
RP/0/0/CPU0:SP2R2#sh rsvp session detail
SESSION: IPv4-LSP Addr: 12.39.5.5, TunID: 25, ExtID: 12.39.2.2
 PSBs: 1, RSBs: 1, Requests: 0
 LSPId: 3
 Tunnel Name: SP2R2_t25
  RSVP Path Info:
   InLabel: No intf, No label
   Incoming Address: Unknown
   Explicit Route:
     Strict, 12.39.23.3/32
     Strict, 12.39.35.5/32
     Strict, 12.39.5.5/32
   Record Route: None
   Tspec: avg rate=50M, burst=1K, peak rate=50M
  RSVP Resv Info:
   OutLabel: GigabitEthernet0/0/0/0, 123307
   FRR OutLabel: No intf, No label
   Record Route: None
   Fspec: avg rate=50M, burst=1K, peak rate=50M
```

Once you remove the admin weight put artificially on Gi0/0/0/2 and resetup the tunnel, it will follow the old path from R2 - to R1 and then to R5:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#mpls traffic-eng
RP/0/0/CPU0:SP2R2(config-mpls-te)#interface gigabitEthernet 0/0/0/2
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#no admin-weight 5000
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#commit
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#
RP/0/0/CPU0:SP2R2#mpls traffic-eng resetup name SP2R2_t25
RP/0/0/CPU0:SP2R2#sh rsvp session detail
SESSION: IPv4-LSP Addr: 12.39.5.5, TunID: 25, ExtID: 12.39.2.2
 PSBs: 1, RSBs: 1, Requests: 0
 LSPId: 4
 Tunnel Name: SP2R2_t25
  RSVP Path Info:
   InLabel: No intf, No label
   Incoming Address: Unknown
   Explicit Route:
     Strict, 12.39.12.1/32
     Strict, 12.39.15.5/32
     Strict, 12.39.5.5/32
   Record Route: None
   Tspec: avg rate=50M, burst=1K, peak rate=50M
  RSVP Resv Info:
   OutLabel: GigabitEthernet0/0/0/2, 123116
   FRR OutLabel: No intf, No label
   Record Route: None
   Fspec: avg rate=50M, burst=1K, peak rate=50M
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3.6: Announcing TE tunnel via forwarding adjacency

Autoroute announce only works within scope of the specific router it was configured on. To announce the tunnel in the IGP, we need to make sure the tunnel is active both ways – from R2 to R5 and back. Otherwise, it will be still visible on the local node, but not advertised within IGP.

So, let's configure matching tunnel on SP2R5 towards SP2R2, with forwarding adjacency:

```
SP2-R5-RR#conf t
SP2-R5-RR(config)#interface tunnel 52
SP2-R5-RR(config-if)#ip unnumbered loopback 0
SP2-R5-RR(config-if)#tunnel mode mpls traffic-eng
SP2-R5-RR(config-if)#tunnel destination 12.39.2.2
SP2-R5-RR(config-if)#tunnel mpls traffic-eng bandwidth 5000
SP2-R5-RR(config-if)#tunnel mpls traffic-eng path-option 10 dynamic
SP2-R5-RR(config-if)#tunnel mpls traffic-eng forwarding-adjacency
SP2-R5-RR(config-if)#^Z
```

And let's change autoroute announce to forwarding adjacency on SP2R2 as well:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#interface tunnel-te25
RP/0/0/CPU0:SP2R2(config-if)#no autoroute announce
RP/0/0/CPU0:SP2R2(config-if)#forwarding-adjacency
RP/0/0/CPU0:SP2R2(config-if-tunte-fwdadj)#commit
```

We can see that the best route to 12.39.2.2 on SP2R5 changed to tunnel 52:

```
SP2-R5-RR#show ip route 12.39.2.2
Routing entry for 12.39.2.2/32
  Known via "isis", distance 115, metric 20, type level-2
  Redistributing via isis AS1239
  Last update from 12.39.2.2 on Tunnel52, 00:00:50 ago
  Routing Descriptor Blocks:
  * 12.39.2.2, from 12.39.2.2, 00:00:50 ago, via Tunnel52
      Route metric is 20, traffic share count is 1
```

The other thing also happened – we have now not two but three same cost paths towards SP2R4 on SP2R5 – two via physical interfaces, and third via tunnel:

```
SP2-R5-RR#show ip route 12.39.4.4
Routing entry for 12.39.4.4/32
  Known via "isis", distance 115, metric 30, type level-2
  Redistributing via isis AS1239
  Last update from 12.39.2.2 on Tunnel52, 00:01:51 ago
  Routing Descriptor Blocks:
    12.39.35.3, from 12.39.4.4, 00:01:51 ago, via GigabitEthernet2
      Route metric is 30, traffic share count is 1
  * 12.39.15.1, from 12.39.4.4, 00:01:51 ago, via GigabitEthernet3
      Route metric is 30, traffic share count is 1
    12.39.2.2, from 12.39.4.4, 00:01:51 ago, via Tunnel52
      Route metric is 30, traffic share count is 1
```

In the real network, such setup would probably be less than desirable – the tunnel is anyway transiting one of those physical links and will influence the bandwidth available.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Let's focus however on the forwarding adjacency. If the tunnel from SP2R2 is advertised in AS1239 IGP, SP2R4 should have best path to SP2R5 via SP2R2, not via SP2R1 and SP2R3 as before. Let's check the route:

```
RP/0/0/CPU0:SP2R4#show ip route 12.39.5.5
Routing entry for 12.39.5.5/32
  Known via "isis AS1239", distance 115, metric 21, type level-2
  Installed Jan  1 21:42:16.079 for 00:04:59
  Routing Descriptor Blocks
    12.39.24.2, from 12.39.5.5, via GigabitEthernet0/0/0/5
      Route metric is 21
  No advertising protos.
```

Notice, that as soon as we'll tear down TE tunnel on SP2R5:

```
SP2-R5-RR#conf t
SP2-R5-RR(config)#no int tu52
SP2-R5-RR(config)#^Z
SP2-R5-RR#
```

The forwarding adjacency will be withdrawn from the IGP. On SP2R4:

```
RP/0/0/CPU0:SP2R4#show ip route 12.39.5.5

Routing entry for 12.39.5.5/32
  Known via "isis AS1239", distance 115, metric 30, type level-2
  Installed Jan  1 21:51:03.893 for 00:00:11
  Routing Descriptor Blocks
    12.39.14.1, from 12.39.5.5, via GigabitEthernet0/0/0/0
      Route metric is 30
    12.39.34.3, from 12.39.5.5, via GigabitEthernet0/0/0/2
      Route metric is 30
  No advertising protos
```

Let's tear down the tunnel, and build the one we've set out to build:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#no int tunnel-te 25
RP/0/0/CPU0:SP2R2(config)#commit
RP/0/0/CPU0:SP2R2(config)#
RP/0/0/CPU0:SP2R2#show mpls forwarding tunnels
RP/0/0/CPU0:SP2R2#
RP/0/0/CPU0:SP2R2#show ip route 12.39.5.5

Routing entry for 12.39.5.5/32
  Known via "isis AS1239", distance 115, metric 30, type level-2
  Installed Jan  1 21:04:15.341 for 00:00:55
  Routing Descriptor Blocks
    12.39.23.3, from 12.39.5.5, via GigabitEthernet0/0/0/0
      Route metric is 30
    12.39.12.1, from 12.39.5.5, via GigabitEthernet0/0/0/2
      Route metric is 30
  No advertising protos.
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4: Building static tunnel

First, we'll define explicit static path of next-hops for the tunnel:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#explicit-path name R2-R3-R4-R1-R5
RP/0/0/CPU0:SP2R2(config-expl-path)#index 10 next-address strict ipv4 unicast 12.39.23.3
RP/0/0/CPU0:SP2R2(config-expl-path)#index 20 next-address strict ipv4 unicast 12.39.34.4
RP/0/0/CPU0:SP2R2(config-expl-path)#index 30 next-address strict ipv4 unicast 12.39.14.1
RP/0/0/CPU0:SP2R2(config-expl-path)#index 40 next-address strict ipv4 unicast 12.39.15.5
RP/0/0/CPU0:SP2R2(config-expl-path)#index 50 next-address strict ipv4 unicast 12.39.5.5
```

Now, let's define the tunnel itself. Now however, with explicit path option:

```
RP/0/0/CPU0:SP2R2(config)#interface tunnel-te 25
RP/0/0/CPU0:SP2R2(config-if)#signalled-bandwidth 5000
RP/0/0/CPU0:SP2R2(config-if)#ipv4 unnumbered loopback 0
RP/0/0/CPU0:SP2R2(config-if)#autoroute announce
RP/0/0/CPU0:SP2R2(config-if-tunte-aa)#metric 1
RP/0/0/CPU0:SP2R2(config-if)#destination 12.39.5.5
RP/0/0/CPU0:SP2R2(config-if)#path-option 10 explicit name R2-R3-R4-R1-R5
RP/0/0/CPU0:SP2R2(config-if)#
```

The tunnel should be up:

```
RP/0/0/CPU0:SP2R2#show int tunnel-te25
tunnel-te25 is up, line protocol is up
  Interface state transitions: 1
[…]
RP/0/0/CPU0:SP2R2#show mpls forwarding tunnels
Tunnel        Outgoing    Outgoing     Next Hop        Bytes
Name          Label       Interface                    Switched
------------- ----------- ------------ --------------- ------------
tt25              123308      Gi0/0/0/0    12.39.23.3      2382
```

Label 123308 tells us, that SP2R3 was responsible for allocation of the label thanks to the label ranges we've did in previous sessions.

Now, let's check traceroute to destination:

```
RP/0/0/CPU0:SP2R2#traceroute 12.39.5.5
Tracing the route to 12.39.5.5
 1  12.39.23.3 [MPLS: Label 123308 Exp 0] 9 msec  9 msec  9 msec
 2  12.39.34.4 [MPLS: Label 123451 Exp 0] 0 msec  9 msec  9 msec
 3  12.39.14.1 [MPLS: Label 123110 Exp 0] 29 msec  9 msec  9 msec
 4  12.39.15.5 9 msec  *  9 msec
```

Let's move to SP2R3 and check the label 123308 detail:

```
RP/0/0/CPU0:SP2R3#show mpls forwarding labels 123308 detail
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
123308 123451      TE: 25            Gi0/0/0/2    12.39.34.4      173171
     Updated: Jan  1 21:19:50.358
     Version: 19, Priority: 2
     Label Stack (Top -> Bottom): { 123451 }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 14/18, MTU: 1500
     Packets Switched: 1177
```

We can see that outgoing label from SP2R3 perspective is 123451, the one we saw in traceroute as next hop (towards R4).

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Running MPLS TE in OSPF (AS 3356)

Task 2.1: Basic IGP and MPLS TE configuration

Please configure MPLS TE, RSVP and OSPF TE extensions on all routers within AS 3356.

On SP1R1-SP1R4:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#router ospf AS3356
RP/0/0/CPU0:SP1R4(config-ospf)#area 0
RP/0/0/CPU0:SP1R4(config-ospf-ar)#mpls traffic-eng
RP/0/0/CPU0:SP1R4(config-ospf-ar)#exit
RP/0/0/CPU0:SP1R4(config-ospf)#mpls traffic-eng router-id loopback 0

RP/0/0/CPU0:SP1R4(config)#mpls traffic-eng
RP/0/0/CPU0:SP1R4(config-mpls-te)#int gi0/0/0/0 ! and on all core-facing interfaces

RP/0/0/CPU0:SP1R4(config)#rsvp
RP/0/0/CPU0:SP1R4(config-rsvp)#int gi0/0/0/0    ! and on all core-facing interfaces
RP/0/0/CPU0:SP1R4(config-rsvp-if)#bandwidth 10 Gbps

RP/0/0/CPU0:SP1R4(config-ospf)#commit
```

On SP1R5:

```
SP1-R5-RR#conf t
SP1-R5-RR(config)#mpls traffic-eng tunnels
SP1-R5-RR(config)#router ospf 100
SP1-R5-RR(config-router)#mpls traffic-eng router-id Loopback0
SP1-R5-RR(config-router)#mpls traffic-eng area 0
SP1-R5-RR(config-router)#^Z
SP1-R5-RR(config)#int gi2
SP1-R5-RR(config-if)#ip rsvp bandwidth 10000000
SP1-R5-RR(config-if)#mpls traffic-eng tunnels
SP1-R5-RR(config-if)#int gi3
SP1-R5-RR(config-if)#ip rsvp bandwidth 10000000
SP1-R5-RR(config-if)#mpls traffic-eng tunnels
SP1-R5-RR(config-if)#^Z
```

To do a basic check of MPLS TE distribution within IGP:

```
RP/0/0/CPU0:SP1R1#show mpls traffic-eng topology | i MPLS TE Id
IGP Id: 33.56.1.1, MPLS TE Id: 33.56.1.1 Router Node  (OSPF AS3356 area 0)
IGP Id: 33.56.2.2, MPLS TE Id: 33.56.2.2 Router Node  (OSPF AS3356 area 0)
IGP Id: 33.56.3.3, MPLS TE Id: 33.56.3.3 Router Node  (OSPF AS3356 area 0)
IGP Id: 33.56.4.4, MPLS TE Id: 33.56.4.4 Router Node  (OSPF AS3356 area 0)
IGP Id: 33.56.5.5, MPLS TE Id: 33.56.5.5 Router Node  (OSPF AS3356 area 0)
```

If there's something wrong, you can filter out the output to just show the TE Ids and the links associated with each node:

```
RP/0/0/CPU0:SP1R1#show mpls traffic-eng topology | i Node
IGP Id: 33.56.1.1, MPLS TE Id: 33.56.1.1 Router Node  (OSPF AS3356 area 0)
  Link[0]:Point-to-Point, Nbr IGP Id:33.56.4.4, Nbr Node Id:4, gen:56
  Link[1]:Point-to-Point, Nbr IGP Id:33.56.2.2, Nbr Node Id:2, gen:57
  Link[2]:Point-to-Point, Nbr IGP Id:33.56.5.5, Nbr Node Id:5, gen:58
  Link[3]:Point-to-Point, Nbr IGP Id:33.56.3.3, Nbr Node Id:3, gen:59
IGP Id: 33.56.2.2, MPLS TE Id: 33.56.2.2 Router Node  (OSPF AS3356 area 0)
  Link[0]:Point-to-Point, Nbr IGP Id:33.56.3.3, Nbr Node Id:3, gen:60
 […]
```

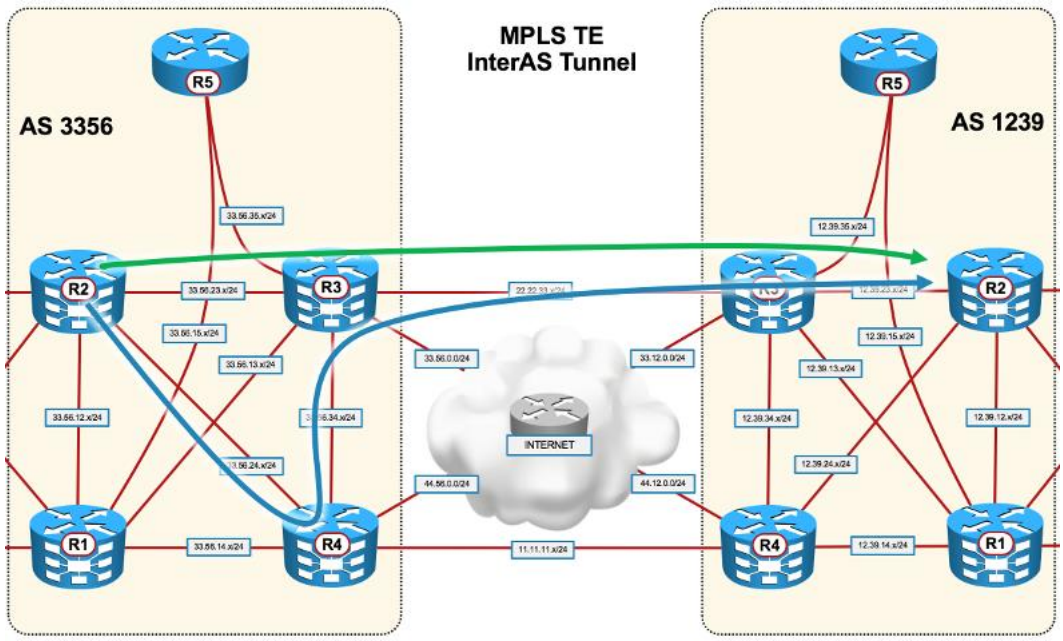

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

All routers within AS3356 should be advertising their own links and presence within the RSVP domain.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.2: Inter-AS MPLS TE Tunnel

Our goal is to build Inter-AS MPLS TE tunnel between SP1R4 and SP2R2:



There is existing label-switched path between SP1R2 and SP2R2 built in previous sessions. Let's trace it over the network:

```
RP/0/0/CPU0:SP1R2#sh mpls forwarding prefix 12.39.2.2/32
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
102028 103037      12.39.2.2/32      Gi0/0/0/0    33.56.23.3      109705

RP/0/0/CPU0:SP1R2#sh route 12.39.2.2
Routing entry for 12.39.2.2/32
  Known via "ospf AS3356", distance 110, metric 1
  Tag 77, type extern 2
  Installed Jan 1 11:23:56.423 for 00:22:03
  Routing Descriptor Blocks
    33.56.23.3, from 33.56.3.3, via GigabitEthernet0/0/0/0
      Route metric is 1
  No advertising protos.
```

Hop to SP1R3:

```
RP/0/0/CPU0:SP1R3#sh mpls forwarding prefix 12.39.2.2/32
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
103037 123302      12.39.2.2/32      Gi0/0/0/1    22.22.33.254    127371
```

Hop to AS1239 on SP2R3:

```
RP/0/0/CPU0:SP2R3#sh mpls forwarding labels 123302
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
123302 Exp-Null-v4 12.39.2.2/32      Gi0/0/0/0    12.39.23.2      1226929
```

…and to SP2R2, from which we see SP1R2, so 33.56.2.2:

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP2R2#sh mpls forwarding prefix 33.56.2.2/32
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
123204 123305      33.56.2.2/32      Gi0/0/0/0    12.39.23.3      253622
```

And we can check the path:

```
RP/0/0/CPU0:SP2R2#traceroute 33.56.2.2

Type escape sequence to abort.
Tracing the route to 33.56.2.2

 1  12.39.23.3 [MPLS: Label 123305 Exp 0] 19 msec  9 msec  0 msec
 2  22.22.33.1 [MPLS: Label 103001 Exp 0] 9 msec  0 msec  0 msec
 3  33.56.23.2 9 msec  *  9 msec
```

The tunnel we're going to build is meant to transport AS100 MPLS L3VPN traffic (vrf 'cust1'). Let's check the path before we make any modifications:

```
RP/0/0/CPU0:SP1R2#traceroute vrf cust1 192.168.7.71

Type escape sequence to abort.
Tracing the route to 192.168.7.71

 1  33.56.23.3 [MPLS: Labels 103037/123220 Exp 0] 9 msec  9 msec  9 msec
 2  22.22.33.254 [MPLS: Labels 123302/123220 Exp 0] 9 msec  9 msec  9 msec
 3  12.39.23.2 [MPLS: Labels 0/123220 Exp 0] 9 msec  9 msec  9 msec
 4  20.40.27.7 9 msec  *  9 msec
```

Now, we'll configure explicit path through AS3356 to SP2R2. Notice you configure only the part within your own Area (OSPF) or Level (IS-IS), then loose expansion will take ownership of steering the tunnel towards defined destination:

```
RP/0/0/CPU0:SP1R2(config)#explicit-path name InterAS-TE
RP/0/0/CPU0:SP1R2(config-expl-path)#index 10 next-address 33.56.24.4
RP/0/0/CPU0:SP1R2(config-expl-path)#index 20 next-address 33.56.34.3
RP/0/0/CPU0:SP1R2(config-expl-path)#index 30 next-address loose 12.39.2.2
RP/0/0/CPU0:SP1R2(config-expl-path)#exit
```

Enable SP1R3 Gi0/0/0/1 and SP2R3 Gi0/0/0/1 interfaces to be used as RSVP path:

```
RP/0/0/CPU0:SP1R3#conf t
RP/0/0/CPU0:SP1R3(config)#mpls traffic-eng
RP/0/0/CPU0:SP1R3(config-mpls-te)#int gi0/0/0/3
RP/0/0/CPU0:SP1R3(config-mpls-te-if)#rsvp
RP/0/0/CPU0:SP1R3(config-rsvp)#int gi0/0/0/3
RP/0/0/CPU0:SP1R3(config-rsvp-if)#bandwidth 10 gbps
RP/0/0/CPU0:SP1R3(config-rsvp-if)#commit
RP/0/0/CPU0:SP1R3(config-rsvp-if)#

RP/0/0/CPU0:SP2R3#conf t
RP/0/0/CPU0:SP2R3(config)#mpls traffic-eng
RP/0/0/CPU0:SP2R3(config-mpls-te)#int gi0/0/0/1
RP/0/0/CPU0:SP2R3(config-mpls-te-if)#rsvp
RP/0/0/CPU0:SP2R3(config-rsvp)#int gi0/0/0/1
RP/0/0/CPU0:SP2R3(config-rsvp-if)#bandwidth 10 gbps
RP/0/0/CPU0:SP2R3(config-rsvp-if)#
```

…and let's build the tunnel using this path:

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R2(config)#int tunnel-te 2432
RP/0/0/CPU0:SP1R2(config-if)#ipv4 unnumbered loopback 0
RP/0/0/CPU0:SP1R2(config-if)#signalled-bandwidth 1000
RP/0/0/CPU0:SP1R2(config-if)#destination 12.39.2.2
RP/0/0/CPU0:SP1R2(config-if)#path-option 10 explicit name InterAS-TE
RP/0/0/CPU0:SP1R2(config-if)#commit
RP/0/0/CPU0:SP1R2(config-if)#
```

Note the autoroute announce won't be effective here

```
RP/0/0/CPU0:SP1R2#sh mpls forwarding prefix 12.39.2.2/32
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
102028 103037      12.39.2.2/32       Gi0/0/0/0    33.56.23.3      339082

RP/0/0/CPU0:SP1R2#traceroute 12.39.2.2

Type escape sequence to abort.
Tracing the route to 12.39.2.2

 1  33.56.23.3 [MPLS: Label 103037 Exp 0] 19 msec  0 msec  9 msec
 2  22.22.33.254 [MPLS: Label 123302 Exp 0] 0 msec  0 msec  9 msec
 3  12.39.23.2 9 msec  *  9 msec
RP/0/0/CPU0:SP1R2#conf t
```

We will need to force the traffic into tunnel using static routing:



The only way to force traffic to Inter-AS TE tunnels right now is either to use static or policy based routing. This comes from the fact that both autoroute and forwarding adjacency features assume that for the head end of tunnel the whole network domain is visible. With Inter-AS only limited information is available to the signalling node and thus it was decided that network operator will need extra effort to put traffic into such tunnel.

```
RP/0/0/CPU0:SP1R2(config)#router static
RP/0/0/CPU0:SP1R2(config-static)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-static-afi)#12.39.2.2/32 tunnel-te 2432
RP/0/0/CPU0:SP1R2(config-static-afi)#commit
RP/0/0/CPU0:SP1R2(config-static-afi)#

RP/0/0/CPU0:SP1R2#sh mpls forwarding prefix 12.39.2.2/32
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
102028 Pop         12.39.2.2/32       tt2432       point2point     0

RP/0/0/CPU0:SP1R2#ping 12.39.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.39.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/11/19 ms

RP/0/0/CPU0:SP1R2#traceroute 12.39.2.2

Type escape sequence to abort.
Tracing the route to 12.39.2.2

 1  33.56.24.4 [MPLS: Label 104035 Exp 0] 9 msec  9 msec  9 msec
 2  33.56.34.3 [MPLS: Label 103011 Exp 0] 9 msec  9 msec  9 msec
 3  22.22.33.254 [MPLS: Label 123307 Exp 0] 9 msec  9 msec  9 msec
 4  12.39.23.2 9 msec  *  9 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

We can check the tunnel details on the headend:

```
RP/0/0/CPU0:SP1R2#sh mpls traffic-eng tunnels

Name: tunnel-te2432  Destination: 12.39.2.2  Ifhandle:0x110
  Signalled-Name: SP1R2_t2432
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type explicit InterAS-TE (Basis for Setup, path weight 1)
        Reroute pending (Explicit-path inuse by the current LSP has been modified)
      Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
      Metric Type: TE (global)
      Path Selection:
        Tiebreaker: Min-fill (default)
      Hop-limit: disabled
      Cost-limit: disabled
      Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    Last PCALC Error [Reopt]: Fri Jan 1 12:29:55 2017
      Info: Can't find destination 0000.0000.0000.00
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 1000 kbps  CT0
    Creation Time: Fri Jan 1 12:06:20 2017 (00:29:18 ago)
  Config Parameters:
    Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  History:
    Tunnel has been up for: 00:07:45 (since Fri Jan 01 12:27:53 UTC 2017)
    Current LSP:
      Uptime: 00:07:45 (since Fri Jan 01 12:27:53 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Fri Jan 01 12:35:32 UTC 2017 [00:00:06 ago]

  Path info (OSPF AS3356 area 0):
  Node hop count: 1
  Hop0: 33.56.24.4
  Hop1: 33.56.4.4
  Hop2: 33.56.3.3
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

Then on ASBR – SP1R3:

```
RP/0/0/CPU0:SP1R3#sh mpls traffic-eng tunnels

LSP Tunnel 33.56.2.2 2432 [30] is signalled, Signaling State: up
  Tunnel Name: SP1R2_t2432 Tunnel Role: Mid
  InLabel: GigabitEthernet0/0/0/2, 103011
  OutLabel: GigabitEthernet0/0/0/1, 123307
  Signalling Info:
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    Src 33.56.2.2 Dst 12.39.2.2, Tun ID 2432, Tun Inst 30, Ext ID 33.56.2.2
    Router-IDs: upstream   33.56.4.4
                local      33.56.3.3
                downstream not required
    Bandwidth: 1000 kbps (CT0) Priority:  7  7 DSTE-class: 0
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Incoming Address: 33.56.34.3
      Incoming:
      Explicit Route:
        Strict, 33.56.34.3
        Strict, 33.56.3.3

      Outgoing:
        No ERO
      Record Route: Disabled
      Tspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
      Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
    Resv Info: None
      Record Route: Disabled
      Fspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
Displayed 0 (of 0) heads, 1 (of 1) midpoints, 0 (of 0) tails
Displayed 0 up, 0 down, 0 recovering, 0 recovered heads
```

And on the ASBR in AS1239 – SP2R3:

```
RP/0/0/CPU0:SP2R3#sh mpls traffic-eng tunnels
LSP Tunnel 33.56.2.2 2432 [30] is signalled, Signaling State: up
  Tunnel Name: SP1R2_t2432 Tunnel Role: Mid
  InLabel: GigabitEthernet0/0/0/1, 123307
  OutLabel: GigabitEthernet0/0/0/0, implicit-null
  Signalling Info:
    Src 33.56.2.2 Dst 12.39.2.2, Tun ID 2432, Tun Inst 30, Ext ID 33.56.2.2
    Router-IDs: upstream   not required
                local      12.39.3.3
                downstream 12.39.2.2
    Bandwidth: 1000 kbps (CT0) Priority:  7  7 DSTE-class: 0
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Incoming Address: 22.22.33.1
      Incoming:
      Explicit Route:
        No ERO
      ERO Expansion Info:
      IS-IS AS1239 level-2, Metric 10 (TE), Affinity 0x0, Mask 0xffff, Queries 1
      Last Better Path Query was received 00:08:58 ago

      Outgoing:
        Explicit Route:
          Strict, 12.39.23.2
          Strict, 12.39.2.2

      Record Route: Disabled
      Tspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
      Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
    Resv Info: None
      Record Route: Disabled
      Fspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
Displayed 0 (of 0) heads, 1 (of 1) midpoints, 0 (of 0) tails
Displayed 0 up, 0 down, 0 recovering, 0 recovered heads
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Now, we can verify that the traffic from AS100 customers within MPLS L3VPN of 'cust1' takes MPLS-TE tunnel path:

```
RP/0/0/CPU0:SP1R2#traceroute vrf cust1 192.168.7.71

Type escape sequence to abort.
Tracing the route to 192.168.7.71

 1  33.56.24.4 [MPLS: Labels 104035/123220 Exp 0] 9 msec  9 msec  9 msec
 2  33.56.34.3 [MPLS: Labels 103011/123220 Exp 0] 9 msec  9 msec  9 msec
 3  22.22.33.254 [MPLS: Labels 123307/123220 Exp 0] 9 msec  9 msec  9 msec
 4  12.39.23.2 [MPLS: Label 123220 Exp 0] 9 msec  9 msec  9 msec
 5  20.40.27.7 9 msec  *  9 msec
```

Notice the 104035 and 103011 labels – those are TE labels as evidenced by asking RSVP and LSD:

```
RP/0/0/CPU0:SP1R2#sh rsvp reservation detail
RESV: IPv4-LSP Session addr: 12.39.2.2. TunID: 2432. LSPId: 30.
 Source addr: 33.56.2.2. ExtID: 33.56.2.2.
 Input adjusted interface: Gi0/0/0/4. Input physical interface: Gi0/0/0/4.
 Next hop: 33.56.24.4 (lih: 0x8).
 Style: Shared-Explicit. Service: Controlled-Load.
 Rate: 1000000 bits/sec. Burst: 1K bytes. Peak: 1M bits/sec.
 MTU min: 40, max: 1500 bytes.
 Flags: None.
 State expires in 341.872 sec.
 Policy:  Accepted. Policy source(s): Default.
 Header info: RSVP TTL=251. IP TTL=251. Flags: 0x1. TOS=0xc0.
 Resource:
  Labels: Outgoing downstream: 104035.

RP/0/0/CPU0:SP1R2#sh mpls lsd forwarding tunnels
Tunnel_Intf, Path_Info: <Type>
tunnel-te2432, (TE-Control), local_lbl=102030, 1 Paths
   1/1: TEv4, 'default':4U, Gi0/0/0/4, nh=33.56.24.4, lbl=104035, tun=tt2432,
weight=0x0, class=0x0 bkup=None mrg_lbl=None, bkup_local_lbl=0, bkup_nh=0.0.0.0
            flags=0x0 ()
```

We can check the label on SP1R4 to trace the MPLS TE path:

```
RP/0/0/CPU0:SP1R4#sh mpls forwarding labels 104035
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
104035 103011      TE: 2432          Gi0/0/0/2    33.56.34.3      68343

RP/0/0/CPU0:SP1R4#sh rsvp reservation detail
RESV: IPv4-LSP Session addr: 12.39.2.2. TunID: 2432. LSPId: 30.
 Source addr: 33.56.2.2. ExtID: 33.56.2.2.
 Input adjusted interface: Gi0/0/0/2. Input physical interface: Gi0/0/0/2.
 Next hop: 33.56.34.3 (lih: 0x6).
 Style: Shared-Explicit. Service: Controlled-Load.
 Rate: 1000000 bits/sec. Burst: 1K bytes. Peak: 1M bits/sec.
 MTU min: 40, max: 1500 bytes.
 Flags: None.
 State expires in 363.090 sec.
 Policy:  Accepted. Policy source(s): Default.
 Header info: RSVP TTL=252. IP TTL=252. Flags: 0x1. TOS=0xc0.
 Resource:
  Labels: Local downstream: 104035, Outgoing downstream: 103011.
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

…and on SP1R3 to finish the tracing within AS3356:

```
RP/0/0/CPU0:SP1R3#sh mpls forwarding labels 103011
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
103011 123307      TE: 2432          Gi0/0/0/1    22.22.33.254    74602

RP/0/0/CPU0:SP1R3#sh rsvp reservation detail
RESV: IPv4-LSP Session addr: 12.39.2.2. TunID: 2432. LSPId: 30.
 Source addr: 33.56.2.2. ExtID: 33.56.2.2.
 Input adjusted interface: Gi0/0/0/1. Input physical interface: Gi0/0/0/1.
 Next hop: 22.22.33.254 (lih: 0x5).
 Style: Shared-Explicit. Service: Controlled-Load.
 Rate: 1000000 bits/sec. Burst: 1K bytes. Peak: 1M bits/sec.
 MTU min: 40, max: 1500 bytes.
 Flags: None.
 State expires in 341.492 sec.
 Policy:  Accepted. Policy source(s): Default.
 Header info: RSVP TTL=253. IP TTL=253. Flags: 0x1. TOS=0xc0.
 Resource:
  Labels: Local downstream: 103011, Outgoing downstream: 123307.
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3: Other MPLS TE features

Task 3.1: MPLS TE setup and holding priority

As you probably noticed, the tunnels are set up by default with both setup and holding priority of 7:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels | i Priori
    Bandwidth:     5000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
```

Priorities run from 0 to 7, with 0 having the highest priority.

First, let's change the signalled bandwidth of TE tunnel 25 in AS1239 to 8Gbps:

```
RP/0/0/CPU0:SP2R2#sh run int tunnel-te 25
interface tunnel-te25
 ipv4 unnumbered Loopback0
 signalled-bandwidth 5000
 autoroute announce
 !
 destination 12.39.5.5
 path-option 10 dynamic
!
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#interface tunnel-te25
RP/0/0/CPU0:SP2R2(config-if)#signalled-bandwidth 8000000
RP/0/0/CPU0:SP2R2(config-if)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

After making the changes, we can see that the bandwidth is now 8Gbps and the last change was refreshed via RSVP:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels 25
Name: tunnel-te25  Destination: 12.39.5.5  Ifhandle:0x110
  Signalled-Name: SP2R2_t25
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
[…]
  Config Parameters:
    Bandwidth:  8000000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
[…]
    Prior LSP:
      ID: 3 Path Option: 10
      Removal Trigger: reoptimization completed
[…]
```

Let's another tunnel to 12.39.5.5 but this time with setup and hold priority of 5 and bandwidth request of 8Gbps:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#interface tunnel-te 251
RP/0/0/CPU0:SP2R2(config-if)#ipv4 unnumbered lo0
RP/0/0/CPU0:SP2R2(config-if)#signalled-bandwidth 8000000
RP/0/0/CPU0:SP2R2(config-if)#autoroute announce
RP/0/0/CPU0:SP2R2(config-if-tunte-aa)#exit
RP/0/0/CPU0:SP2R2(config-if)#destination 12.39.5.5
RP/0/0/CPU0:SP2R2(config-if)#path-option 10 dynamic
RP/0/0/CPU0:SP2R2(config-if)#priority 5 5
RP/0/0/CPU0:SP2R2(config-if)#commit
RP/0/0/CPU0:SP2R2(config-if)#
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

We can check the status of our two tunnels:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels
Name: tunnel-te25  Destination: 12.39.5.5  Ifhandle:0x110
  Signalled-Name: SP2R2_t25
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type dynamic  (Basis for Setup, path weight 20)
    Last Signalled Error : Wed Jan  1 16:42:53 2017
      Info: [4] PathErr(2,5)-(Admin, flow preempted) at 12.39.15.1

    ! notice the flow was preempted at SP2R1 by another flow

    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 8000000 kbps  CT0
    Creation Time: Wed Jan  1 11:04:47 2017 (19:38:10 ago)
[…]
  History:
    Tunnel has been up for: 00:00:04 (since Tue Jan 01 16:42:53 UTC 2017)
    Current LSP:
      Uptime: 00:00:04 (since Tue Jan 01 16:42:53 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, identical to the [CURRENT] LSP
        Date/Time: Tue Jan 01 00:04:53 UTC 2017 [16:38:04 ago]
    Prior LSP:
      ID: 4 Path Option: 10
      Removal Trigger: path error
Path info (IS-IS AS1239 level-2):
  Node hop count: 2
  Hop0: 12.39.23.3 ! R2->R3
  Hop1: 12.39.35.5 ! R3->R5
  Hop2: 12.39.5.5  ! R5 GE interface -> R5 loopback

Name: tunnel-te251  Destination: 12.39.5.5  Ifhandle:0x130
  Signalled-Name: SP2R2_t251
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    ! our new tunnel to 12.39.5.5

[…]

  Path info (IS-IS AS1239 level-2):
  Node hop count: 2
  Hop0: 12.39.12.1 ! R2->R1
  Hop1: 12.39.15.5 ! R1->R5
  Hop2: 12.39.5.5  ! R5 GE interface -> R5 loopback
```

So right now, from SP2R2 perspective, we have two TE tunnels and we can actually load-balance traffic on them:

```
RP/0/0/CPU0:SP2R2#sh route ipv4 unicast 12.39.5.5

Routing entry for 12.39.5.5/32
  Known via "isis AS1239", distance 115, metric 30, type level-2
  Installed Jan  1 16:42:54.147 for 00:06:49
  Routing Descriptor Blocks
    12.39.5.5, from 12.39.5.5, via tunnel-te25
      Route metric is 30
    12.39.5.5, from 12.39.5.5, via tunnel-te251
      Route metric is 30
  No advertising protos.
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

We can verify the details on the labels assigned:

```
RP/0/0/CPU0:SP2R2#sh mpls forwarding prefix 12.39.5.5/32 detail
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
123203 Pop         12.39.5.5/32       tt25         12.39.5.5       0
     Updated: Jan  1 16:42:54.187
     Version: 72, Priority: 3
     Label Stack (Top -> Bottom): { 123308 Imp-Null }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 14/18, MTU: 1500
     Packets Switched: 0

       Pop         12.39.5.5/32       tt251        12.39.5.5       0
     Updated: Jan  1 16:42:54.187
     Version: 72, Priority: 3
     Label Stack (Top -> Bottom): { 123107 Imp-Null }
     NHID: 0x0, Encap-ID: N/A, Path idx: 1, Backup path idx: 0, Weight: 0
     MAC/Encaps: 14/18, MTU: 1500
     Packets Switched: 0
```

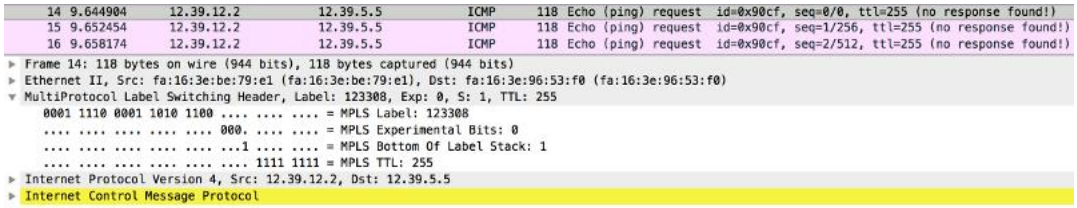
CEF confirms that, notice however, CEF doesn't know anything about outgoing labels:

```
RP/0/0/CPU0:SP2R2#sh cef 12.39.5.5
12.39.5.5/32, version 72, internal 0x1000001 0x0 (ptr 0xa1422874) [1], 0x0 (0xa13ee328),
0xa20 (0xa18fd050)
  Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 12.39.5.5/32, tunnel-te25, 3 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0xa10e434c 0xa10e4790]
    next hop 12.39.5.5/32
    local adjacency
     local label 123203      labels imposed {ImplNull}
   via 12.39.5.5/32, tunnel-te251, 3 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 1 NHID 0x0 [0xa10e4ad8 0xa10e4a30]
    next hop 12.39.5.5/32
    local adjacency
     local label 123203      labels imposed {ImplNull}
```

RSVP signals the labels for both tunnels:

```
RP/0/0/CPU0:SP2R2#sh mpls lsd forwarding tunnels
Tunnel_Intf, Path_Info: <Type>
tunnel-te25, (TE-Control), local_lbl=123221, 1 Paths
   1/1: TEv4, 'default':4U, Gi0/0/0/0, nh=12.39.23.3, lbl=123308, tun=tt25, weight=0x0,
class=0x0 bkup=None mrg_lbl=None, bkup_local_lbl=0, bkup_nh=0.0.0.0
            flags=0x0 ()
tunnel-te251, (TE-Control), local_lbl=123211, 1 Paths
   1/1: TEv4, 'default':4U, Gi0/0/0/2, nh=12.39.12.1, lbl=123107, tun=tt251, weight=0x0,
class=0x0 bkup=None mrg_lbl=None, bkup_local_lbl=0, bkup_nh=0.0.0.0
            flags=0x0 ()
```

And we can verify that indeed, for TE tunnel 25, the outgoing label is 123308 (notice the capture doesn't see ICMP ping responses, they're coming back over the other tunnel):



© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Also, on both next hops, we can verify the label allocation that was advertised via RSVP towards head-end for the tunnel reservation.

On SP2R3:

```
RP/0/0/CPU0:SP2R3#sh mpls forwarding labels 123308
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
123308 Pop         TE: 25             Gi0/0/0/3    12.39.35.5      85000
```

On SP2R1:

```
RP/0/0/CPU0:SP2R1#sh mpls forwarding labels 123107
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
123107 Pop         TE: 251            Gi0/0/0/3    12.39.15.5      20960
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2: RSVP authentication

RSVP can be authenticated globally, per neighbor (TE Id) or per interface. To configure global authentication with key-chain do on SP1R2 (tunnel headend):

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#key chain KEY-RSVP
RP/0/0/CPU0:SP1R2(config-KEY-RSVP)# key 100
RP/0/0/CPU0:SP1R2(config-KEY-RSVP-100)#key-string RSVP
RP/0/0/CPU0:SP1R2(config-KEY-RSVP-100)#accept-lifetime 00:00:00 january 01 2017
RP/0/0/CPU0:SP1R2(config-KEY-RSVP-100)#send-lifetime 00:00:00 january 01 2017
RP/0/0/CPU0:SP1R2(config-KEY-RSVP-100)#cryptographic-algorithm HMAC-MD5
RP/0/0/CPU0:SP1R2(config-KEY-RSVP-100)#accept-tolerance infinite
RP/0/0/CPU0:SP1R2(config-KEY-RSVP-100)#rsvp
RP/0/0/CPU0:SP1R2(config-rsvp)#authentication
RP/0/0/CPU0:SP1R2(config-rsvp-auth)#key-source key-chain KEY-RSVP
RP/0/0/CPU0:SP1R2(config-rsvp-auth)#window-size 16
RP/0/0/CPU0:SP1R2(config-rsvp-auth)#commit
RP/0/0/CPU0:SP1R2#
```

We can see RSVP authentication view:

```
RP/0/0/CPU0:SP1R2#sh rsvp authentication

Codes: S - static, G - global, N - neighbor, I -interface, C - chain

Source Address  Dest Address    Interface   Mode Key-Source Key-ID Code
33.56.24.2      12.39.2.2       Gi0/0/0/4   Send KEY-RSVP   100    SGC
33.56.24.2      33.56.24.4      Gi0/0/0/4   Send KEY-RSVP   100    SGC
33.56.4.4       33.56.24.2      Gi0/0/0/4   Recv KEY-RSVP   100    SGC
33.56.24.4      33.56.24.2      Gi0/0/0/4   Recv KEY-RSVP   100    SGC
```

And the RSVP trace should show authentication information being attached per neighbor:

```
Jan 1 17:08:12.168 rsvp/unique 0/0/CPU0 2# t1  AUTH:813: Created Rx SA for src
(33.56.4.4) dst (33.56.24.2) nbr (33.56.4.4) GigabitEthernet0/0/0/4 (ifh 0xc0)
Jan 1 17:09:11.164 rsvp/unique 0/0/CPU0 2# t1  AUTH:1656: Global KI update: update flags
0x8, window_size 16, lifetime 0
Jan 1 17:08:12.118 rsvp/sig 0/0/CPU0 t1  AUTH:813: Created Rx SA for src (33.56.24.4)
dst (33.56.24.2) nbr (33.56.24.4) GigabitEthernet0/0/0/4 (ifh 0xc0)
Jan 1 17:08:12.168 rsvp/sig 0/0/CPU0 t1  AUTH:813: Created Rx SA for src (33.56.4.4) dst
(33.56.24.2) nbr (33.56.4.4) GigabitEthernet0/0/0/4 (ifh 0xc0)
Jan 1 17:09:11.164 rsvp/sig 0/0/CPU0 t1  AUTH:1656: Global KI update: update flags 0x1,
window_size 0, lifetime 0
Jan 1 17:09:11.164 rsvp/sig 0/0/CPU0 t1  AUTH:1656: Global KI update: update flags 0x8,
window_size 16, lifetime 0
```

Remove RSVP authentication before proceeding.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.3: Naming MPLS-TE tunnels

Tunnels by default have derived name from interface identifier. For example, for tunnel defined with ‘interface tunnel-te 25’, the name will be ‘tunnel-te25’:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels | i Name
Name: tunnel-te25  Destination: 12.39.5.5  Ifhandle:0xb0
  Signalled-Name: SP2R2_t25
```

In real life, with a lot of tunnels, that kind of naming can be problematic however. There’s option to explicitly define the name for tunnel:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#interface tunnel-te 25
RP/0/0/CPU0:SP2R2(config-if)#signalled-name ?
  WORD  The name to be included in signaling (max 254 characters)
RP/0/0/CPU0:SP2R2(config-if)#signalled-name MySP-BCPL-R2-R5
RP/0/0/CPU0:SP2R2(config-if)#commit
```

Now, let’s check the name:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels | i Name
Name: tunnel-te25  Destination: 12.39.5.5  Ifhandle:0xb0
  Signalled-Name: MySP-BCPL-R2-R5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.4: Named MPLS-TE tunnels

In newer IOS XR releases, there's an option to define named MPLS TE right from the start. It's done under 'mpls traffic-eng' configuration option and assumes You'll want both name the tunnel, and name path options. It'll be preferred option going forward with the introduction of other options and features (including, but not limited to, Segment Routing). Configuration is almost the same as with regular MPLS-TE tunnel, but there's no option to specify path options by index (but by name) and no option to define signalled name for the tunnel – as the name is already defined. Let's define simple named MPLS TE tunnel in 'new' configuration model:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#mpls traffic-eng
RP/0/0/CPU0:SP2R2(config-mpls-te)#named-tunnels ?
  tunnel-te  Traffic Engineering Tunnel
  <cr>
RP/0/0/CPU0:SP2R2(config-mpls-te)#named-tunnels tunnel-te ?
  WORD  Tunnel Name
RP/0/0/CPU0:SP2R2(config-mpls-te)#named-tunnels tunnel-te MyTunnel-SPBCPL-1
RP/0/0/CPU0:SP2R2(config-mpls-te-tunnel-name)#destination 12.39.5.5
RP/0/0/CPU0:SP2R2(config-mpls-te-tunnel-name)#signalled-bandwidth 1500
RP/0/0/CPU0:SP2R2(config-mpls-te-tunnel-name)#path-option ?
  WORD  Path setup name ! no option to define path option by index
RP/0/0/CPU0:SP2R2(config-mpls-te-tunnel-name)#path-option PathForNamedTunnel1
RP/0/0/CPU0:SP2R2(config-path-option-name)#preference 10
  ! preference for specific path option
RP/0/0/CPU0:SP2R2(config-path-option-name)#computation dynamic
RP/0/0/CPU0:SP2R2(config-path-option-name)#commit
RP/0/0/CPU0:SP2R2#
```

The tunnel will be existing next to indexed MPLS TE tunnels, as shown below:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels brief

                      TUNNEL NAME          DESTINATION      STATUS  STATE
                      tunnel-te25            12.39.5.5        down  down
                MyTunnel-SPBCPL-1            12.39.5.5        down  down
Displayed 2 (of 2) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 0 up, 2 down, 0 recovering, 0 recovered heads
```

The tunnel itself is also properly signalled (with name) and with specified path option (for the example, don't worry that it's 'down' – it's for the sake of example only):

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels name MyTunnel-SPBCPL-1


Name: MyTunnel-SPBCPL-1  Destination: 12.39.5.5  Ifhandle:0xd0
  Tunnel-ID: 32768
  Status:
    Admin:    up Oper: down   Path: not valid   Signalling: Down

    path option PathForNamedTunnel1, preference 10,  type dynamic
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Mon Jan 01 13:30:29 2017 (00:05:59 ago)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.5: FlexLSP – bidirectional tunnels

As setting up bidirectional tunnels that follow the same path between routers is sometimes challenging with bigger topologies, IOS XR offers feature to build co-routed MPLS TE tunnels (that feature is also available in IOS-XE starting in 16.5.1b). Let's do a simple co-routed MPLS TE tunnel between R2 and R4 in AS3356. To enable something more complex than just direct connection between the routers, shut down Gi0/0/0/4 on R2 leading to R4 before you start the configuration.

MPLS TE tunnel is defined as usual, but you need to configure the bidirectional feature. For that, there's an local ID needed (global is optional) and decision where the IP source address for tunnel identifier will be located. In this example we'll use R2 loopback0 interface.

Tunnel definition on R2:

```
SP1R2#conf t
SP1R2(config)#interface tunnel-te 234
SP1R2(config-if)#ipv4 unnumbered loopback 0
SP1R2(config-if)#destination 33.56.4.4
SP1R2(config-if)#path-option 10 dynamic
SP1R2(config-if)#bidirectional
SP1R2(config-if-bidir)#association type co-routed
SP1R2(config-if-bidir)#association id 10 source-address 33.56.2.2 global-id 1000
```

Tunnel definition on R4:

```
interface tunnel-te432
 ipv4 unnumbered Loopback0
 destination 33.56.2.2
 bidirectional
  association id 10 source-address 33.56.2.2 global-id 1000 ! notice the source address
  association type co-routed
  !
 !
 path-option 10 dynamic
!
```

Now, the tunnels will be created from both sides and in addition – we get the bonus of them following the same path (co-routed):

As You can check (if everything went OK), on R2, we have two tunnels – we act as one headend and as one tail end:

```
RP/0/0/CPU0:SP1R2#sh mpls traffic-eng tunnels brief

                      TUNNEL NAME          DESTINATION      STATUS  STATE
                     tunnel-te234            33.56.4.4          up  up
                       SP1R4_t432            33.56.2.2          up  up
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 1 (of 1) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

The MPLS TE tunnel output will however vary a bit from the typical tunnel – it'll reflect the fact that it's a co-routed bidirectional one:

```
RP/0/0/CPU0:SP1R2#sh mpls traffic-eng tunnels 234

Name: tunnel-te234  Destination: 33.56.4.4  Ifhandle:0xb0
  Signalled-Name: SP1R2_t234
  Status:
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type dynamic  (Basis for Setup, path weight 2 (reverse 2))
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Mon Jan 0114:08:39 2017 (00:30:40 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Not Enabled
    Association Type: Single Sided Bidirectional LSPs, Co-routed: YES
    Association ID: 10, Source: 33.56.2.2, Global ID: 1000
    Reverse Bandwidth: 0 kbps (CT0), Standby: 0 kbps (CT0)
    LSP Wrap Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  History:
    Tunnel has been up for: 00:12:27 (since Mon Jan 0114:26:52 UTC 2017)
    Current LSP:
      Uptime: 00:12:27 (since Mon Jan 0114:26:52 UTC 2017)
    Prior LSP:
      ID: 7 Path Option: 10
      Removal Trigger: path tear

  Path info (OSPF AS3356 area 0):
  Hop0: 33.56.12.1
  Hop1: 33.56.14.4
  Hop2: 33.56.4.4
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 1) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

As you can see, our tunnel goes through R1 to R4 from R2 and vice versa. What we'd like to do now is to explicitly define the path to go over R3 rather than R1. For that, it's enough to change the 'dynamic' path-option on R2 to explicit in this way:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#explicit-path name R2-R3-R4
RP/0/0/CPU0:SP1R2(config-expl-path)#index 10 next-address ipv4 unicast 33.56.3.3
RP/0/0/CPU0:SP1R2(config-expl-path)#index 20 next-address ipv4 unicast 33.56.4.4
RP/0/0/CPU0:SP1R2(config-expl-path)#commit
RP/0/0/CPU0:SP1R2#sh explicit-paths
Path R2-R3-R4    status enabled
        10:  next-address strict 33.56.3.3
        20:  next-address strict 33.56.4.4
```

Now, let's change the option on the tunnel:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#interface tunnel-te 234
RP/0/0/CPU0:SP1R2(config-if)#no path-option 10
RP/0/0/CPU0:SP1R2(config-if)#path-option 10 explicit name R2-R3-R4
RP/0/0/CPU0:SP1R2(config-if)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

…and now, let's recheck how the tunnel right now is constructed:

```
RP/0/0/CPU0:SP1R2#sh mpls traffic-eng tunnels 234

Name: tunnel-te234  Destination: 33.56.4.4  Ifhandle:0xb0
  Signalled-Name: SP1R2_t234
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type explicit R2-R3-R4 (Basis for Setup, path weight 2 (reverse 2))
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Mon Jan 0114:08:39 2017 (00:38:38 ago)
  Config Parameters:
[…]
    Association Type: Single Sided Bidirectional LSPs, Co-routed: YES
    Association ID: 10, Source: 33.56.2.2, Global ID: 1000
[…]
  History:
    Tunnel has been up for: 00:03:51 (since Mon Jan 0114:43:26 UTC 2017)
    Current LSP:
      Uptime: 00:03:51 (since Mon Jan 0114:43:26 UTC 2017)
    Prior LSP:
      ID: 7 Path Option: 10
      Removal Trigger: path tear

  Path info (OSPF AS3356 area 0):
  Node hop count: 2
  Hop0: 33.56.23.3  ! R2->R3
  Hop1: 33.56.34.4  ! R3->R4
  Hop2: 33.56.4.4   ! R4
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 1) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

And let's check the tunnel from R4:

```
RP/0/0/CPU0:SP1R4#sh mpls traffic-eng tunnels 432

Name: tunnel-te432  Destination: 33.56.2.2  Ifhandle:0xb0
  Signalled-Name: SP1R4_t432
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type dynamic  (Basis for Setup, path weight 2 (reverse 2))
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Mon Jan 0114:13:38 2017 (00:36:08 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Not Enabled
    Association Type: Single Sided Bidirectional LSPs, Co-routed: YES
    Association ID: 10, Source: 33.56.2.2, Global ID: 1000
    Reverse Bandwidth: 0 kbps (CT0), Standby: 0 kbps (CT0)
    LSP Wrap Protection: Not Enabled
    BFD Fast Detection: Disabled
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  History:
    Tunnel has been up for: 00:06:21 (since Mon Jan 0114:43:25 UTC 2017)
    Current LSP:
      Uptime: 00:06:21 (since Mon Jan 0114:43:25 UTC 2017)
    Prior LSP:
      ID: 14 Path Option: 10
      Removal Trigger: path verification failed

  Path info (OSPF AS3356 area 0):
  Hop0: 33.56.34.3
  Hop1: 33.56.23.2
  Hop2: 33.56.2.2
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 1) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.5.1: Co-routed bidirectional tunnels: autoroute

The tunnel itself won't carry any traffic without additional configuration. Let's enable autoroute announce on it to check how we see R4 from R2. Before:

```
RP/0/0/CPU0:SP1R2#sh ip route
[…]
O    33.56.4.4/32 [110/3] via 33.56.23.3, 00:23:52, GigabitEthernet0/0/0/0
                  [110/3] via 33.56.12.1, 00:23:52, GigabitEthernet0/0/0/2
```

Adding autoroute announce:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#int tunnel-te 234
RP/0/0/CPU0:SP1R2(config-if)#autoroute announce
RP/0/0/CPU0:SP1R2(config-if-tunte-aa)#commit
```

And now let's check the route again:

```
RP/0/0/CPU0:SP1R2#sh ip route 33.56.4.4

Routing entry for 33.56.4.4/32
  Known via "ospf AS3356", distance 110, metric 3, type intra area
  Installed Jan 01 14:52:11.523 for 00:02:16
  Routing Descriptor Blocks
    33.56.4.4, from 33.56.4.4, via tunnel-te234
      Route metric is 3
  No advertising protos.
```

Verification:

```
RP/0/0/CPU0:SP1R2#traceroute 33.56.4.4

Type escape sequence to abort.
Tracing the route to 33.56.4.4

 1  33.56.23.3 [MPLS: Label 103004 Exp 0] 9 msec  0 msec  0 msec
 2  33.56.34.4 0 msec  *  0 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.5.2: Co-routed bidirectional tunnels: forwarding adjacency

The other option to forward traffic would be to configure FAs on both ends. Let's do that:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#int tunnel-te234
RP/0/0/CPU0:SP1R2(config-if)#no autoroute announce
RP/0/0/CPU0:SP1R2(config-if)#forwarding-adjacency
RP/0/0/CPU0:SP1R2(config-if-tunte-fwdadj)#commit
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#int tunnel-te 432
RP/0/0/CPU0:SP1R4(config-if)#forwarding-adjacency
RP/0/0/CPU0:SP1R4(config-if-tunte-fwdadj)#commit
```

Let's verify it:

```
RP/0/0/CPU0:SP1R2#sh mpls traffic-eng forwarding-adjacency
destination 33.56.4.4 has 1 tunnels

             tunnel-te234    (traffic share 0, next-hop 33.56.4.4)
                             (Adjacency Announced: yes, holdtime 1)
                             (OSPF AS3356 area 0, IPV4 Unicast)

RP/0/0/CPU0:SP1R2#sh route 33.56.4.4

Routing entry for 33.56.4.4/32
  Known via "ospf AS3356", distance 110, metric 2, type intra area
  Installed Jan 28 15:26:12.853 for 00:06:15
  Routing Descriptor Blocks
    directly connected, via tunnel-te234
      Route metric is 2
  No advertising protos.
```

And on R4:

```
RP/0/0/CPU0:SP1R4#sh route 33.56.2.2

Routing entry for 33.56.2.2/32
  Known via "ospf AS3356", distance 110, metric 2, type intra area
  Installed Jan 28 15:26:12.044 for 00:08:28
  Routing Descriptor Blocks
    directly connected, via tunnel-te432
      Route metric is 2
  No advertising protos.

RP/0/0/CPU0:SP1R4#sh mpls traffic-eng forwarding-adjacency
 destination 33.56.2.2 has 1 tunnels

             tunnel-te432    (traffic share 0, next-hop 33.56.2.2)
                             (Adjacency Announced: yes, holdtime 1)
                             (OSPF AS3356 area 0, IPV4 Unicast)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4: MPLS TE path protection

Task 4.1: Building protected path from SP2R2 to SP2R5

Let's tear down TE 251 tunnel on SP2R2:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#no int tunnel-te 251
RP/0/0/CPU0:SP2R2(config)#commit
RP/0/0/CPU0:SP2R2(config)#
```

Now, we're going to request path protection for TE tunnel 25:

```
RP/0/0/CPU0:SP2R2(config)#int tunnel-te25
RP/0/0/CPU0:SP2R2(config-if)#path-protection
RP/0/0/CPU0:SP2R2(config-if)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

Let's see what changed in the tunnel information:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels

Name: tunnel-te25  Destination: 12.39.5.5  Ifhandle:0x110
  Signalled-Name: SP2R2_t25
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type dynamic  (Basis for Setup, path weight 20)
    Last Signalled Error : Tue Jan  1 16:42:53 2017
      Info: [4] PathErr(2,5)-(Admin, flow preempted) at 12.39.15.1
    path option 10,  type dynamic  (Basis for Standby, path weight 20)
[…]
  Config Parameters:
[…]
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Enabled
[…]
  History:
    Tunnel has been up for: 00:29:12 (since Tue Jan 01 16:42:53 UTC 2017)
    Current LSP:
      Uptime: 00:29:12 (since Tue Jan 01 16:42:53 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, identical to the [CURRENT] LSP
        Date/Time: Tue Jan 01 17:04:53 UTC 2017 [00:07:12 ago]
    Prior LSP:
      ID: 4 Path Option: 10
      Removal Trigger: path error
    Standby LSP:
      Uptime: 00:00:32 (since Tue Jan 01 17:11:33 UTC 2017)

  Path info (IS-IS AS1239 level-2):
  Node hop count: 2
  Hop0: 12.39.23.3
  Hop1: 12.39.35.5
  Hop2: 12.39.5.5

  Standby LSP Path info (IS-IS AS1239 level-2), Oper State: Up :
  Node hop count: 2
  Hop0: 12.39.12.1
  Hop1: 12.39.15.5
  Hop2: 12.39.5.5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Notice however, that right now we're using twice the reservation in the network – once via SP2R1 and once via SP2R3, as both LSPs allocate bandwidth:

```
RP/0/0/CPU0:SP2R3#sh mpls traffic-eng tunnels

LSP Tunnel 12.39.2.2 25 [5] is signalled, Signaling State: up
  Tunnel Name: SP2R2_t25 Tunnel Role: Mid
  InLabel: GigabitEthernet0/0/0/0, 123308
  OutLabel: GigabitEthernet0/0/0/3, implicit-null
  Signalling Info:
    Src 12.39.2.2 Dst 12.39.5.5, Tun ID 25, Tun Inst 5, Ext ID 12.39.2.2
    Router-IDs: upstream   12.39.2.2
                local      12.39.3.3
                downstream 12.39.5.5
    Bandwidth: 8000000 kbps (CT0) Priority:  7  7 DSTE-class: 0
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Incoming Address: 12.39.23.3
      Incoming:
      Explicit Route:
        Strict, 12.39.23.3
        Strict, 12.39.35.5
        Strict, 12.39.5.5

      Outgoing:
        Explicit Route:
          Strict, 12.39.35.5
          Strict, 12.39.5.5

      Record Route: Disabled
      Tspec: avg rate=8000000 kbits, burst=1000 bytes, peak rate=8000000 kbits
      Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
    Resv Info: None
      Record Route: Disabled
      Fspec: avg rate=8000000 kbits, burst=1000 bytes, peak rate=8000000 kbits

RP/0/0/CPU0:SP2R3#sh rsvp int

*: RDM: Default I/F B/W % : 75% [default] (max resv/bc0), 0% [default] (bc1)

Interface                MaxBW (bps)  MaxFlow (bps) Allocated (bps)      MaxSub (bps)
------------------------ ------------ ------------- -------------------- -------------
GigabitEthernet0/0/0/0            10G           10G             1M (  0%)             0
GigabitEthernet0/0/0/1            10G           10G              0 (  0%)             0
GigabitEthernet0/0/0/2            10G           10G              0 (  0%)             0
GigabitEthernet0/0/0/3            10G           10G             8G ( 80%)             0
GigabitEthernet0/0/0/6            10G           10G              0 (  0%)             0
```

And on SP2R1:

```
RP/0/0/CPU0:SP2R1#sh mpls traffic-eng tunnels

LSP Tunnel 12.39.2.2 25 [6] is signalled, Signaling State: up
  Tunnel Name: SP2R2_t25 Tunnel Role: Mid
  InLabel: GigabitEthernet0/0/0/2, 123106
  OutLabel: GigabitEthernet0/0/0/3, implicit-null
  Signalling Info:
    Src 12.39.2.2 Dst 12.39.5.5, Tun ID 25, Tun Inst 6, Ext ID 12.39.2.2
    Router-IDs: upstream   12.39.2.2
                local      12.39.1.1
                downstream 12.39.5.5
    Bandwidth: 8000000 kbps (CT0) Priority:  7  7 DSTE-class: 0
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Incoming Address: 12.39.12.1
      Incoming:
      Explicit Route:
        Strict, 12.39.12.1
        Strict, 12.39.15.5
        Strict, 12.39.5.5

      Outgoing:
        Explicit Route:
          Strict, 12.39.15.5
          Strict, 12.39.5.5

      Record Route: Disabled
      Tspec: avg rate=8000000 kbits, burst=1000 bytes, peak rate=8000000 kbits
      Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
    Resv Info: None
      Record Route: Disabled
      Fspec: avg rate=8000000 kbits, burst=1000 bytes, peak rate=8000000 kbits
Displayed 0 (of 0) heads, 1 (of 1) midpoints, 0 (of 0) tails
Displayed 0 up, 0 down, 0 recovering, 0 recovered heads
RP/0/0/CPU0:SP2R1#sh rsvp int

*: RDM: Default I/F B/W % : 75% [default] (max resv/bc0), 0% [default] (bc1)

Interface                MaxBW (bps)  MaxFlow (bps) Allocated (bps)      MaxSub (bps)
------------------------ ------------ ------------- -------------------- -------------
GigabitEthernet0/0/0/0            10G            10G              0 (  0%)             0
GigabitEthernet0/0/0/2            10G            10G              0 (  0%)             0
GigabitEthernet0/0/0/3            10G            10G             8G ( 80%)             0
GigabitEthernet0/0/0/4            10G            10G              0 (  0%)             0
```

We will now fail one path and check the tunnel status:

```
RP/0/0/CPU0:SP2R3#conf t
RP/0/0/CPU0:SP2R3(config)#int gi0/0/0/3
RP/0/0/CPU0:SP2R3(config-if)#shut
RP/0/0/CPU0:SP2R3(config-if)#commit
RP/0/0/CPU0:SP2R3(config-if)#
RP/0/0/CPU0:SP2R3#sh rsvp int

*: RDM: Default I/F B/W % : 75% [default] (max resv/bc0), 0% [default] (bc1)

Interface                MaxBW (bps)  MaxFlow (bps) Allocated (bps)      MaxSub (bps)
------------------------ ------------ ------------- -------------------- -------------
GigabitEthernet0/0/0/0            10G            10G             1M (  0%)             0
GigabitEthernet0/0/0/1            10G            10G              0 (  0%)             0
GigabitEthernet0/0/0/2            10G            10G              0 (  0%)             0
GigabitEthernet0/0/0/3            10G            10G              0 (  0%)             0
GigabitEthernet0/0/0/6            10G            10G              0 (  0%)             0
```

So, on SP2R2 right now:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels 25

Name: tunnel-te25  Destination: 12.39.5.5  Ifhandle:0x110
  Signalled-Name: SP2R2_t25
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type dynamic  (Basis for Setup, path weight 20)
    Last PCALC Error [Standby]: Wed Jan  1 17:27:39 2017
      Info: No diverse path found
    Last Signalled Error : Wed Jan  1 17:27:35 2017
      Info: [5] PathErr(24,5)-(routing, no route to dest) at 12.39.23.3
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 8000000 kbps  CT0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    Creation Time: Tue Jan  1 21:04:47 2017 (20:23:26 ago)
  Config Parameters:
    Bandwidth:  8000000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  History:
    Tunnel has been up for: 00:45:20 (since Tue Jan 01 16:42:53 UTC 2017)
    Current LSP:
      Uptime: 00:16:40 (since Tue Jan 01 17:11:33 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, identical to the [CURRENT] LSP
        Date/Time: Tue Jan 01 17:04:53 UTC 2017 [00:23:20 ago]
    Prior LSP:
      ID: 5 Path Option: 10
      Removal Trigger: path error

  Path info (IS-IS AS1239 level-2):
  Node hop count: 2
  Hop0: 12.39.12.1
  Hop1: 12.39.15.5
  Hop2: 12.39.5.5
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 1) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

We see that the protected path is now primary one. We can also verify, that the SP1R1 advertised label will be used for protection:

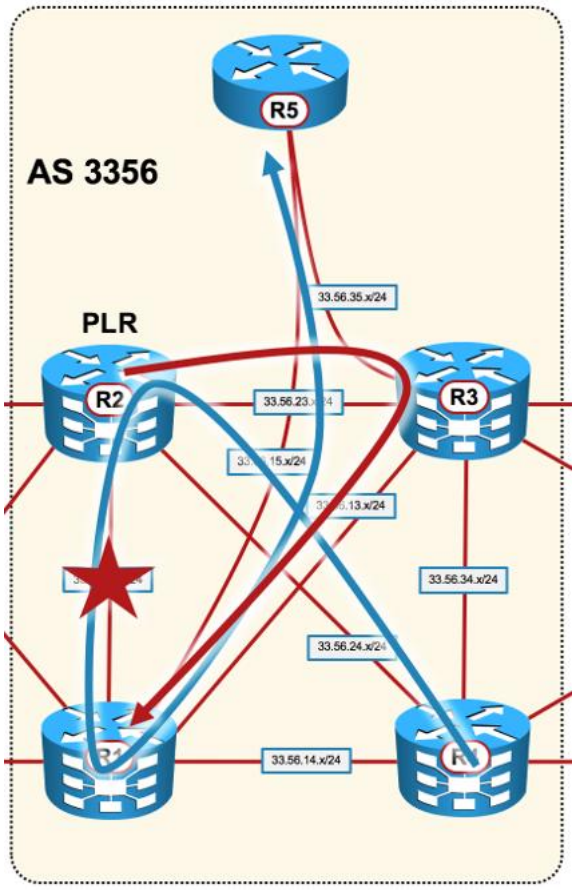
```
RP/0/0/CPU0:SP2R2#sh mpls forwarding labels 123203 detail
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
123203 Pop         12.39.5.5/32      tt25         12.39.5.5       0
     Updated: Jan  1 16:42:54.187
     Version: 75, Priority: 3
     Label Stack (Top -> Bottom): { 123106 Imp-Null }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 14/18, MTU: 1500
     Packets Switched: 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5: MPLS TE link protection

Task 5.1: Configuring MPLS TE link protection (NHOP)

To showcase the fast reroute capabilities, we'll go with following configuration within AS3356:



R4 is setting TE tunnel to R5 going via R2, R1 and R5. Blue line is primary tunnel path, and with red we have our protection path.

We're concerned about R2<>R1 link, and we want to protect the traffic going over it – so R2 will be our PLR (Point of Local Repair). So, on SP1R4:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#explicit-path name R4-R5
RP/0/0/CPU0:SP1R4(config-expl-path)#index 10 next-address 33.56.24.2
RP/0/0/CPU0:SP1R4(config-expl-path)#index 20 next-address 33.56.12.1
RP/0/0/CPU0:SP1R4(config-expl-path)#index 30 next-address 33.56.15.5
RP/0/0/CPU0:SP1R4(config-expl-path)#index 40 next-address 33.56.5.5
RP/0/0/CPU0:SP1R4(config-expl-path)#exit
RP/0/0/CPU0:SP1R4(config)#inte tunnel-te 45
RP/0/0/CPU0:SP1R4(config-if)#ipv4 unnumbered loopback 0
RP/0/0/CPU0:SP1R4(config-if)#signalled-bandwidth 1000
RP/0/0/CPU0:SP1R4(config-if)#autoroute announce
RP/0/0/CPU0:SP1R4(config-if-tunte-aa)#exit
RP/0/0/CPU0:SP1R4(config-if)#destination 33.56.5.5
RP/0/0/CPU0:SP1R4(config-if)#path-option 10 explicit name R4-R5
RP/0/0/CPU0:SP1R4(config-if)#path-option 20 dynamic
RP/0/0/CPU0:SP1R4(config-if)#fast-reroute
RP/0/0/CPU0:SP1R4(config-if)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Let's verify:

```
RP/0/0/CPU0:SP1R4#sh mpls traffic-eng tunnels
Name: tunnel-te45  Destination: 33.56.5.5  Ifhandle:0x110
  Signalled-Name: SP1R4_t45
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type explicit R4-R5 (Basis for Setup, path weight 3)
    path option 20,  type dynamic
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 1000 kbps  CT0
    Creation Time: Mon Jan  1 19:25:55 2017 (00:00:06 ago)
  Config Parameters:
    Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Enabled, Protection Desired: Any
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  History:
    Tunnel has been up for: 00:00:05 (since Mon Jan  1 19:25:56 UTC 2017)
    Current LSP:
      Uptime: 00:00:05 (since Mon Jan  1 19:25:56 UTC 2017)
    Prior LSP:
      ID: 2 Path Option: 10
      Removal Trigger: path tear

  Path info (OSPF AS3356 area 0):
  Node hop count: 3
  Hop0: 33.56.24.2
  Hop1: 33.56.12.1
  Hop2: 33.56.15.5
  Hop3: 33.56.5.5
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads

RP/0/0/CPU0:SP1R4#sh route

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

Gateway of last resort is not set

C    10.0.128.0/29 is directly connected, 3d19h, GigabitEthernet0/0/0/3
L    10.0.128.2/32 is directly connected, 3d19h, GigabitEthernet0/0/0/3
C    11.11.11.0/24 is directly connected, 3d19h, GigabitEthernet0/0/0/1
L    11.11.11.1/32 is directly connected, 3d19h, GigabitEthernet0/0/0/1
S    11.11.11.254/32 is directly connected, 3d19h, GigabitEthernet0/0/0/1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
B    12.39.0.0/16 [20/0] via 11.11.11.254, 2d21h
O E2 12.39.2.2/32 [110/1] via 33.56.34.3, 00:05:09, GigabitEthernet0/0/0/2
O    22.22.33.0/24 [110/2] via 33.56.34.3, 1d20h, GigabitEthernet0/0/0/2
B    33.56.0.0/16 [200/0] via 0.0.0.0, 3d19h, Null0
O    33.56.1.1/32 [110/2] via 33.56.14.1, 3d19h, GigabitEthernet0/0/0/0
O    33.56.2.2/32 [110/2] via 33.56.24.2, 3d19h, GigabitEthernet0/0/0/5
O    33.56.3.3/32 [110/2] via 33.56.34.3, 3d19h, GigabitEthernet0/0/0/2
L    33.56.4.4/32 is directly connected, 3d19h, Loopback0
O    33.56.5.5/32 [110/3] via 33.56.5.5, 00:00:10, tunnel-te45
O    33.56.12.0/24 [110/2] via 33.56.14.1, 3d19h, GigabitEthernet0/0/0/0
                   [110/2] via 33.56.24.2, 3d19h, GigabitEthernet0/0/0/5
O    33.56.13.0/24 [110/2] via 33.56.14.1, 3d19h, GigabitEthernet0/0/0/0
                   [110/2] via 33.56.34.3, 3d19h, GigabitEthernet0/0/0/2
RP/0/0/CPU0:SP1R4#traceroute 33.56.5.5

Type escape sequence to abort.
Tracing the route to 33.56.5.5

 1  33.56.24.2 [MPLS: Label 102031 Exp 0] 9 msec  9 msec  0 msec
 2  33.56.12.1 [MPLS: Label 101009 Exp 0] 0 msec  0 msec  0 msec
 3  33.56.15.5 0 msec  *  9 msec
```

Let's move on to R2 and configure backup tunnel there:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#explicit-path name R2-R1-via-R3
RP/0/0/CPU0:SP1R2(config-expl-path)#index 10 next-address 33.56.3.3
RP/0/0/CPU0:SP1R2(config-expl-path)#index 20 next-address 33.56.1.1
RP/0/0/CPU0:SP1R2(config-expl-path)#exit
RP/0/0/CPU0:SP1R2(config)#int tunnel-te 231
RP/0/0/CPU0:SP1R2(config-if)#ipv4 unnumbered loopback 0
RP/0/0/CPU0:SP1R2(config-if)#signalled-bandwidth 1000
RP/0/0/CPU0:SP1R2(config-if)#destination 33.56.1.1
RP/0/0/CPU0:SP1R2(config-if)#path-option 10 explicit name R2-R1-via-R3
RP/0/0/CPU0:SP1R2(config)#mpls traffic-eng
RP/0/0/CPU0:SP1R2(config-mpls-te)#interface gigabitEthernet 0/0/0/2
RP/0/0/CPU0:SP1R2(config-mpls-te-if)#backup-path tunnel-te 231
RP/0/0/CPU0:SP1R2(config-mpls-te-if)#commit
RP/0/0/CPU0:SP1R2(config-mpls-te-if)#
RP/0/0/CPU0:SP1R2#
```

On R2 we should have our own tunnel up and be the midpoint for the normal operation of the R4->R5 one:

```
RP/0/0/CPU0:SP1R2#sh mpls traffic-eng tunnels brief

                  TUNNEL NAME          DESTINATION      STATUS  STATE
                 tunnel-te231            33.56.1.1          up  up
                    SP1R4_t45            33.56.5.5          up  up
Displayed 1 (of 1) heads, 1 (of 1) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

However, we also have FRR database with one entry thanks to the configuration of backup tunnel for Gi0/0/0/2:

```
RP/0/0/CPU0:SP1R2#sh mpls traffic-eng fast-reroute database
LSP midpoint FRR information:
LSP Identifier                Local Label Out Intf : Label   FRR Intf : Label   Status
----------------------------- ----------- ------------------ ------------------ --------
33.56.4.4 45 [5]              102031      Gi0/0/0/2:101011   tt231:101011       Ready
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

So let's now simulate network problem by shutting down gi0/0/0/2 on R2:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#int gi0/0/0/2
RP/0/0/CPU0:SP1R2(config-if)#shut
RP/0/0/CPU0:SP1R2(config-if)#commit
```

We can see that FRR kicked on:

```
RP/0/0/CPU0:SP1R2#sh mpls traffic-eng fast-reroute database
LSP midpoint FRR information:
LSP Identifier                Local Label Out Intf : Label   FRR Intf : Label   Status
----------------------------- ----------- ------------------ ------------------ -------
33.56.4.4 45 [5]              102031      tt231:101011                          Active
```

On SP1R4 the change is visible:

```
RP/0/0/CPU0:SP1R4#sh mpls traffic-eng tunnels

Name: tunnel-te45  Destination: 33.56.5.5  Ifhandle:0x110
  Signalled-Name: SP1R4_t45
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type explicit R4-R5 (Basis for Setup, path weight 3)
        Change in required resources detected: reroute pending
      Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
      Metric Type: TE (global)
      Path Selection:
        Tiebreaker: Min-fill (default)
      Hop-limit: disabled
      Cost-limit: disabled
      Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    Last PCALC Error [Reopt]: Tue Jan  1 19:45:57 2017
      Info: Explicit path has unknown address, 33.56.12.1
    Last Signalled Error : Tue Jan  1 19:45:22 2017
      Info: [5] PathErr(25,3)-(notify, local repair) at 33.56.24.2
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 1000 kbps  CT0
    Creation Time: Tue Jan  1 19:25:55 2017 (00:20:14 ago)
  Config Parameters:
    Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Enabled, Protection Desired: Any
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  History:
    Tunnel has been up for: 00:06:42 (since Tue Jan 01 19:39:27 UTC 2017)
    Current LSP:
      Uptime: 00:06:42 (since Tue Jan 01 19:39:27 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Tue Jan 01 19:45:57 UTC 2017 [00:00:12 ago]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    Prior LSP:
      ID: 5 Path Option: 10
      Removal Trigger: path error

  Path info (OSPF AS3356 area 0):
  Node hop count: 3
  Hop0: 33.56.24.2
  Hop1: 33.56.12.1
  Hop2: 33.56.15.5
  Hop3: 33.56.5.5
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

And we can doublecheck this with traceroute to 33.56.5.5:

```
RP/0/0/CPU0:SP1R4#traceroute 33.56.5.5

Type escape sequence to abort.
Tracing the route to 33.56.5.5

 1  33.56.24.2 [MPLS: Label 102031 Exp 0] 9 msec  9 msec  0 msec
 2  33.56.23.3 [MPLS: Labels 103012/101011 Exp 0] 0 msec  0 msec  0 msec
 3  33.56.13.1 [MPLS: Label 101011 Exp 0] 0 msec  9 msec  9 msec
 4  33.56.15.5 9 msec  *  9 msec
```

Notice the two labels on 33.56.23.3 hop: 103012 and 101011. Going hop by hop, on R2:

```
RP/0/0/CPU0:SP1R2#sh mpls forwarding prefix 33.56.1.1/32
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
102001 Pop         33.56.1.1/32       tt231        33.56.1.1       1808
RP/0/0/CPU0:SP1R2#sh mpls lsd forwarding tunnels
Tunnel_Intf, Path_Info: <Type>
tunnel-te231, (TE-Control), local_lbl=102032, 1 Paths
   1/1: TEv4, 'default':4U, Gi0/0/0/0, nh=33.56.23.3, lbl=103012, tun=tt231, weight=0x0,
class=0x0 bkup=None mrg_lbl=None, bkup_local_lbl=0, bkup_nh=0.0.0.0
            flags=0x0 ()
```

On R3:

```
RP/0/0/CPU0:SP1R3#sh mpls forwarding labels 103012
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
103012 Pop         TE: 231            Gi0/0/0/6    33.56.13.1      9451
```

And finally on R1:

```
RP/0/0/CPU0:SP1R1#sh mpls forwarding labels 101011
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
101011 Pop         TE: 45             Gi0/0/0/3    33.56.15.5      9062
```

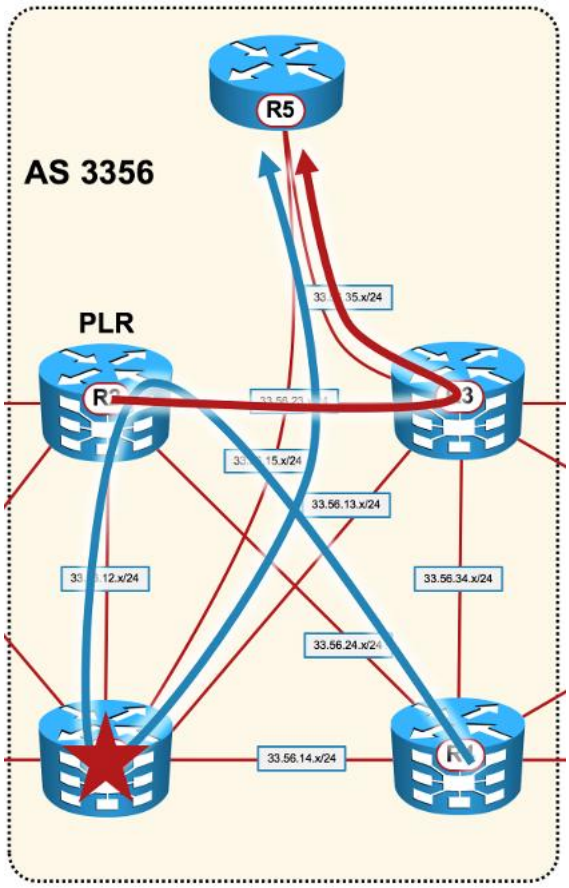
Remove tunnel on R2, unshut Gi0/0/0/2 interface and unconfigure backup-path of Gi0/0/0/2 in MPLS TE configuration section before proceeding.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 6: MPLS TE node protection

Task 6.1: Configuring MPLS TE node protection (NNHOP)

We will protect our tunnel from R4 with the node protection – we want to avoid problems if R1 goes away:



We'll start with the R4->R5 tunnel configuration from previous task:

```
RP/0/0/CPU0:SP1R4#sh run int tunnel-te 45
interface tunnel-te45
 ipv4 unnumbered Loopback0
 signalled-bandwidth 1000
 autoroute announce
 !
 destination 33.56.5.5
 fast-reroute
 path-option 10 explicit name R4-R5
!
```

But we'll add request to FRR to provide node protection:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#interface tunnel-te 45
RP/0/0/CPU0:SP1R4(config-if)#fast-reroute protect node
RP/0/0/CPU0:SP1R4(config-if)#
```

Before we go further, let's check the status of TE tunnel on R4:

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R4#sh mpls traffic-eng tunnels detail
Name: tunnel-te45  Destination: 33.56.5.5  Ifhandle:0x110
  Signalled-Name: SP1R4_t45
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type explicit R4-R5 (Basis for Setup, path weight 3)
    Last Signalled Error : Tue Jan  1 19:45:22 2017
      Info: [5] PathErr(25,3)-(notify, local repair) at 33.56.24.2
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 1000 kbps  CT0
    Creation Time: Tue Jan  1 19:25:55 2017 (00:46:14 ago)
  Config Parameters:
    Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Enabled, Protection Desired: Node
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  SNMP Index: 18
  Binding SID: None
  History:
    Tunnel has been up for: 00:12:33 (since Tue Jan 01 19:59:36 UTC 2017)
    Current LSP:
      Uptime: 00:12:33 (since Tue Jan 01 19:59:36 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Tue Jan 01 19:52:09 UTC 2017 [00:20:00 ago]
    Prior LSP:
      ID: 5 Path Option: 10
      Removal Trigger: resv delete
  Current LSP Info:
    Instance: 6, Signaling Area: OSPF AS3356 area 0
    Uptime: 00:12:33 (since Tue Jan 01 19:59:36 UTC 2017)
    Outgoing Interface: GigabitEthernet0/0/0/5, Outgoing Label: 102031
    Router-IDs: local      33.56.4.4
                downstream 33.56.2.2
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Outgoing:
        Explicit Route:
          Strict, 33.56.24.2
          Strict, 33.56.12.1
          Strict, 33.56.15.5
          Strict, 33.56.5.5

      Record Route: Disabled
      Tspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
      Session Attributes: Local Prot: Set, Node Prot: Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
    Resv Info:
      Record Route:
        IPv4 33.56.2.2, flags 0x20 (Node-ID)
        Label 102031, flags 0x1
        IPv4 33.56.24.2, flags 0x0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
        Label 102031, flags 0x1
        IPv4 33.56.1.1, flags 0x20 (Node-ID)
        Label 101009, flags 0x1
        IPv4 33.56.12.1, flags 0x0
        Label 101009, flags 0x1
        IPv4 33.56.5.5, flags 0x20 (Node-ID)
        Label 3, flags 0x1
      Fspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
  Persistent Forwarding Statistics:
    Out Bytes: 14571
    Out Packets: 224
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

Now we need to add the capability on R2 to protect the path:

```
RP/0/0/CPU0:SP1R2(config)#explicit-path name R2-R5
RP/0/0/CPU0:SP1R2(config-expl-path)#index 10 exclude-address 33.56.1.1
RP/0/0/CPU0:SP1R2(config-expl-path)#exit
RP/0/0/CPU0:SP1R2(config)#interface tunnel-te 25
RP/0/0/CPU0:SP1R2(config-if)#ipv4 unnumbered loopback 0
RP/0/0/CPU0:SP1R2(config-if)#signalled-bandwidth 1000
RP/0/0/CPU0:SP1R2(config-if)#destination 33.56.5.5
RP/0/0/CPU0:SP1R2(config-if)#path-option 10 explicit name  R2-R5
RP/0/0/CPU0:SP1R2(config-if)#commit
```

…and the backup interface configuration:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#mpls traffic-eng
RP/0/0/CPU0:SP1R2(config-mpls-te)#int gi0/0/0/2
RP/0/0/CPU0:SP1R2(config-mpls-te-if)#backup-path tunnel-te 25
RP/0/0/CPU0:SP1R2(config-mpls-te-if)#commit

RP/0/0/CPU0:SP1R2#sh mpls traffic-eng fast-reroute database
LSP midpoint FRR information:
LSP Identifier                Local Label Out Intf : Label   FRR Intf : Label   Status
----------------------------- ----------- ------------------ ------------------ -------
33.56.4.4 45 [6]              102031      Gi0/0/0/2:101009   tt25:Pop           Ready
```

We can check with RSVP how the current situation looks like:

```
RP/0/0/CPU0:SP1R2#show rsvp reservation detail
RESV: IPv4-LSP Session addr: 33.56.5.5. TunID: 25. LSPId: 2.
 Source addr: 33.56.2.2. ExtID: 33.56.2.2.
 Input adjusted interface: Gi0/0/0/0. Input physical interface: Gi0/0/0/0.
 Next hop: 33.56.23.3 (lih: 0x4).
 Style: Shared-Explicit. Service: Controlled-Load.
 Rate: 1000000 bits/sec. Burst: 1K bytes. Peak: 1M bits/sec.
 MTU min: 0, max: 1500 bytes.
 Flags: None.
 State expires in 351.541 sec.
 Policy:  Accepted. Policy source(s): Default.
 Header info: RSVP TTL=253. IP TTL=254. Flags: 0x1. TOS=0xc0.
 Resource:
  Labels: Outgoing downstream: 103011.

RESV: IPv4-LSP Session addr: 33.56.5.5. TunID: 45. LSPId: 6.
 Source addr: 33.56.4.4. ExtID: 33.56.4.4.
 Input adjusted interface: Gi0/0/0/2. Input physical interface: Gi0/0/0/2.
 Next hop: 33.56.12.1 (lih: 0x6).
 Style: Shared-Explicit. Service: Controlled-Load.
 Rate: 1000000 bits/sec. Burst: 1K bytes. Peak: 1M bits/sec.
 MTU min: 0, max: 1500 bytes.
 Flags: None.
 State expires in 348.081 sec.
 Policy:  Accepted. Policy source(s): Default.
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
 Header info: RSVP TTL=253. IP TTL=253. Flags: 0x1. TOS=0xc0.
 Resource:
  Labels: Local downstream: 102031, Outgoing downstream: 101009, Merge-Pt: 3.
```

And back on R4 let's check now the TE tunnel capabilities:

```
RP/0/0/CPU0:SP1R4#sh mpls traffic-eng tunnels role head detail

Name: tunnel-te45  Destination: 33.56.5.5  Ifhandle:0x110
  Signalled-Name: SP1R4_t45
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type explicit R4-R5 (Basis for Setup, path weight 3)
    Last Signalled Error : Tue Jan  1 19:45:22 2017
      Info: [5] PathErr(25,3)-(notify, local repair) at 33.56.24.2
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 1000 kbps  CT0
    Creation Time: Tue Jan  1 19:25:55 2017 (00:55:58 ago)
  Config Parameters:
    Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Enabled, Protection Desired: Node
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  SNMP Index: 18
  Binding SID: None
  History:
    Tunnel has been up for: 00:22:17 (since Tue Jan 01 19:59:36 UTC 2017)
    Current LSP:
      Uptime: 00:22:17 (since Tue Jan 01 19:59:36 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Tue Jan 01 19:52:09 UTC 2017 [00:29:44 ago]
    Prior LSP:
      ID: 5 Path Option: 10
      Removal Trigger: resv delete
  Current LSP Info:
    Instance: 6, Signaling Area: OSPF AS3356 area 0
    Uptime: 00:22:17 (since Tue Jan 01 19:59:36 UTC 2017)
    Outgoing Interface: GigabitEthernet0/0/0/5, Outgoing Label: 102031
    Router-IDs: local      33.56.4.4
                downstream 33.56.2.2
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Outgoing:
        Explicit Route:
          Strict, 33.56.24.2
          Strict, 33.56.12.1
          Strict, 33.56.15.5
          Strict, 33.56.5.5

      Record Route: Disabled
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Tspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
      Session Attributes: Local Prot: Set, Node Prot: Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
    Resv Info:
      Record Route:
        IPv4 33.56.2.2, flags 0x29 (Node-ID, Protection: available, node)
        Label 102031, flags 0x1
        IPv4 33.56.24.2, flags 0x9 (Protection: available, node)
        Label 102031, flags 0x1
        IPv4 33.56.1.1, flags 0x20 (Node-ID)
        Label 101009, flags 0x1
        IPv4 33.56.12.1, flags 0x0
        Label 101009, flags 0x1
        IPv4 33.56.5.5, flags 0x20 (Node-ID)
        Label 3, flags 0x1
      Fspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
  Persistent Forwarding Statistics:
    Out Bytes: 17738
    Out Packets: 266
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

Now, again, let's shut down Gi0/0/0/2 interface on R2 to simulate R1 failure:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#int gi0/0/0/2
RP/0/0/CPU0:SP1R2(config-if)#shut
RP/0/0/CPU0:SP1R2(config-if)#commit
RP/0/0/CPU0:SP1R2(config-if)#
```

We see FRR kicked on:

```
RP/0/0/CPU0:SP1R2#sh mpls traffic-eng fast-reroute database
LSP midpoint FRR information:
LSP Identifier                Local Label Out Intf : Label   FRR Intf : Label   Status
----------------------------- ----------- ------------------ ------------------ -------
33.56.4.4 45 [6]              102031      tt25:Pop                              Active
```

So let's check that on R4:

```
RP/0/0/CPU0:SP1R4#sh mpls traffic-eng tunnels role head detail

Name: tunnel-te45  Destination: 33.56.5.5  Ifhandle:0x110
  Signalled-Name: SP1R4_t45
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type explicit R4-R5 (Basis for Setup, path weight 3)
        Change in required resources detected: reroute pending
      Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
      Metric Type: TE (global)
      Path Selection:
        Tiebreaker: Min-fill (default)
      Hop-limit: disabled
      Cost-limit: disabled
      Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    Last PCALC Error [Reopt]: Tue Jan  1 20:23:22 2017
      Info: Explicit path has unknown address, 33.56.12.1
    Last Signalled Error : Tue Jan  1 20:22:44 2017
      Info: [6] PathErr(25,3)-(notify, local repair) at 33.56.24.2
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 1000 kbps  CT0
    Creation Time: Tue Jan  1 19:25:55 2017 (00:58:16 ago)
  Config Parameters:
    Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Enabled, Protection Desired: Node
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  SNMP Index: 18
  Binding SID: None
  History:
    Tunnel has been up for: 00:24:35 (since Tue Jan 01 19:59:36 UTC 2017)
    Current LSP:
      Uptime: 00:24:35 (since Tue Jan 01 19:59:36 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Tue Jan 01 20:23:41 UTC 2017 [00:00:30 ago]
    Prior LSP:
      ID: 6 Path Option: 10
      Removal Trigger: path error
  Current LSP Info:
    Instance: 6, Signaling Area: OSPF AS3356 area 0
    Uptime: 00:24:35 (since Tue Jan 01 19:59:36 UTC 2017)
    Outgoing Interface: GigabitEthernet0/0/0/5, Outgoing Label: 102031
    Router-IDs: local      33.56.4.4
                downstream 33.56.2.2
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Outgoing:
        Explicit Route:
          Strict, 33.56.24.2
          Strict, 33.56.12.1
          Strict, 33.56.15.5
          Strict, 33.56.5.5

      Record Route: Disabled
      Tspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
      Session Attributes: Local Prot: Set, Node Prot: Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
    Resv Info:
      Record Route:
        IPv4 33.56.2.2, flags 0x29 (Node-ID, Protection: available, node)
        Label 102031, flags 0x1
        IPv4 33.56.24.2, flags 0x9 (Protection: available, node)
        Label 102031, flags 0x1
        IPv4 33.56.1.1, flags 0x20 (Node-ID)
        Label 101009, flags 0x1
        IPv4 33.56.12.1, flags 0x0
        Label 101009, flags 0x1
        IPv4 33.56.5.5, flags 0x20 (Node-ID)
        Label 3, flags 0x1
      Fspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
  Persistent Forwarding Statistics:
    Out Bytes: 18319
    Out Packets: 274
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

And traceroute to 33.56.5.5:

```
RP/0/0/CPU0:SP1R4#traceroute 33.56.5.5

Type escape sequence to abort.
Tracing the route to 33.56.5.5

 1   33.56.24.2 [MPLS: Label 102031 Exp 0] 9 msec  0 msec  0 msec
 2   33.56.23.3 [MPLS: Label 103011 Exp 0] 0 msec  0 msec  0 msec
 3   33.56.35.5 0 msec  *  0 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 7: MPLS TE advanced path selection

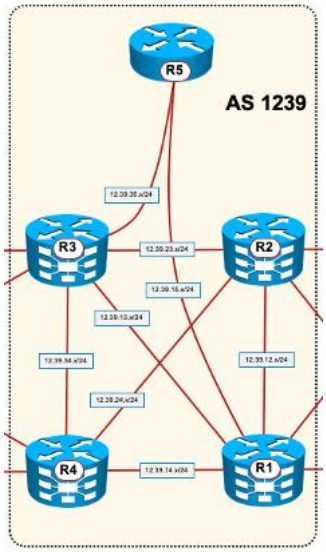
Task 7.1: Link affinity with new configuration style - IOS XR

RFC 3630 (for OSPFv2) and RFC 5305 (for IS-IS) define Administrative Groups TE link attribute. This enables flexible mapping of which links should be included and which excluded from CSPF calculations for TE tunnels.



Before continuing, please remove all TE tunnels in AS 1239 for them to not interfere with our exercises.

Lets assume two links: R2<>R1 and R2<>R3 are physically transported using one DWDM link and they often cause problems in our deployment. So if one of those links fails, second will fail as well. We don't want to build tunnels between R2 and R5 using those links, as link between R2 and R4 is "safer". We can achieve this goal using various means, but one of those is link affinity definition.



On SP2R2 lets define AG called DWDM1 and assign it to both Gi0/0/0/2 as well as Gi0/0/0/0 (interfaces leading to SP2R1 and SP2R3):

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#mpls traffic-eng
RP/0/0/CPU0:SP2R2(config-mpls-te)#interface gigabitEthernet 0/0/0/0
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#attribute-names DWDM1
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#interface gigabitEthernet 0/0/0/2
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#attribute-names DWDM1
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#exit
RP/0/0/CPU0:SP2R2(config-mpls-te)#affinity-map DWDM1 bit-position 31
RP/0/0/CPU0:SP2R2(config-mpls-te)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The name is mapped to specific bitfield index using the **'affinity-map'** command, and then announced using IGP (IS-IS in this case) to all other routers.



We selected position 0 in 4-octet TLV (totally arbitrarily!), and this is "safe" standarized well in the industry. You can use extended range (EAG attribute), but this will not be interoperable between different vendors as of now (for the lab however, you're free to use any range as you won't find any other vendor devices):

```
RP/0/0/CPU0:SP1R2(config-mpls-te)#affinity-map DWDM2 bit-position ?
  <0-255>  Bit Position
```

There's important reason why we selected bit position 31 (last one in 4-octet space) vs position 0 or 1 - default value/mask pair for affinity values is 0x00000000/0x0000ffff which means values from 0 to 15 are by default "masked" (if mask value is 1, then the affinity bit should be set to 0 to match). We'll cover this later.

We can take a look into IS-IS database to check if those affinities are distributed across IGP domain - for R2:

```
RP/0/0/CPU0:SP2R2#sh isis mpls traffic-eng adv | utility egrep "IS-IS|Link|Affinity"
IS-IS AS1239 Level-2 MPLS Traffic Engineering advertisements
  System ID: SP2R2
  Link Count: 3
    Link[0]
      Neighbor System ID: SP2R3 (P2P link)
      Affinity Bits: 0x80000000
    Link[1]
      Neighbor System ID: SP2R1 (P2P link)
      Affinity Bits: 0x80000000
    Link[2]
      Neighbor System ID: SP2R4 (P2P link)
      Affinity Bits: 0x00000000
```

Deciphering bit positions to existing names is hard, but you can ask directly TE database for mapped names:

```
RP/0/0/CPU0:SP2R2#show mpls traffic-eng topology 12.39.2.2 | utility egrep "Link|Attrib"
  Link[0]:Point-to-Point, Nbr IGP Id:0000.0000.0001.00, Nbr Node Id:6, gen:23896
      Attribute Flags: 0x80000000
      Attribute Names: DWDM1(31)
  Link[1]:Point-to-Point, Nbr IGP Id:0000.0000.0004.00, Nbr Node Id:9, gen:23897
      Attribute Flags: 0x0
      Attribute Names:
  Link[2]:Point-to-Point, Nbr IGP Id:0000.0000.0003.00, Nbr Node Id:8, gen:23898
      Attribute Flags: 0x80000000
      Attribute Names: DWDM1(31)
```

The same info can be queried from other routers in the MPLS TE IGP domain, for example on SP2R1:

```
RP/0/0/CPU0:SP2R1# show mpls traffic-eng topology 12.39.2.2 | utility egrep
"Link|Attrib"
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Link[0]:Point-to-Point, Nbr IGP Id:0000.0000.0001.00, Nbr Node Id:1, gen:23889
      Attribute Flags: 0x80000000
      Attribute Names:
  Link[1]:Point-to-Point, Nbr IGP Id:0000.0000.0004.00, Nbr Node Id:4, gen:23890
      Attribute Flags: 0x0
      Attribute Names:
  Link[2]:Point-to-Point, Nbr IGP Id:0000.0000.0003.00, Nbr Node Id:3, gen:23891
      Attribute Flags: 0x80000000
      Attribute Names:
```

Note however, that the names are not transported over IGP TLVs.

Now, lets build another tunnel to R5, but this time explicitly forbid it to take any link with DWDM1 attribute flag:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#interface tunnel-te251
RP/0/0/CPU0:SP2R2(config-if)#ipv4 unnumbered Loopback0
RP/0/0/CPU0:SP2R2(config-if)#signalled-bandwidth 50000
RP/0/0/CPU0:SP2R2(config-if)#autoroute announce
RP/0/0/CPU0:SP2R2(config-if-tunte-aa)#destination 12.39.5.5
RP/0/0/CPU0:SP2R2(config-if)#affinity exclude DWDM1
RP/0/0/CPU0:SP2R2(config-if)#path-option 10 dynamic
RP/0/0/CPU0:SP2R2(config-if)#commit
RP/0/0/CPU0:SP2R2(config-if)#
```

We should have now two tunnels - one via R1 to R5, and then the other using only available link - to R4 and then to R5, excluding interfaces marked with attribute group 31 (named as DWDM1):

```
RP/0/0/CPU0:SP2R2#sh rsvp session detail

SESSION: IPv4-LSP Addr: 12.39.5.5, TunID: 25, ExtID: 12.39.2.2
 PSBs: 1, RSBs: 1, Requests: 0
 LSPId: 7
 Tunnel Name: SP2R2_t25
[...]
   Explicit Route:
     Strict, 12.39.12.1/32
     Strict, 12.39.15.5/32
     Strict, 12.39.5.5/32
[...]

SESSION: IPv4-LSP Addr: 12.39.5.5, TunID: 251, ExtID: 12.39.2.2
 PSBs: 1, RSBs: 1, Requests: 0
 LSPId: 2
 Tunnel Name: SP2R2_t251
[..]
   Explicit Route:
     Strict, 12.39.24.4/32
     Strict, 12.39.34.3/32
     Strict, 12.39.35.5/32
     Strict, 12.39.5.5/32
[...]
```

MPLS TE tunnel information will contain more detailed information about affinity values:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels 251

Name: tunnel-te251  Destination: 12.39.5.5  Ifhandle:0xd0
  Signalled-Name: SP2R2_t251
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    path option 10,  type dynamic  (Basis for Setup, path weight 30)
[...]
  Config Parameters:
    Bandwidth:    50000 kbps (CT0) Priority:  7  7
    Number of affinity constraints: 1
       Exclude bit map           : 0x80000000
       Exclude ext bit map       :
           Length: 256 bits
           Value : 0x::8000:0
       Exclude affinity name     : DWDM1(31)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 7.2: Link affinity with old configuration style - IOS XR

Old style configuration would have the same effect on traffic steering capabilities, but requires more cumbersome preparation.

For old style configuration, you'd need to match interface-level attribute-flags with TE tunnel requirements of affinity and mask. Remember that this needs to match on all path (all links) taken into consideration to build a complete tunnel, not only on the head-end router link towards first potential next-hop.

Example similar to that from previous task to test for yourself would be to have following configuration (after removing new-style configuration that is):

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#interface tunnel-te251
RP/0/0/CPU0:SP2R2(config-if)#affinity 0x00000000 mask 0xffffffff
 ! tunnel will use ONLY links where the affinity value is 0 and all bits match
RP/0/0/CPU0:SP2R2(config-if)#exit
RP/0/0/CPU0:SP2R2(config)#mpls traffic-eng
RP/0/0/CPU0:SP2R2(config-mpls-te)#int gi0/0/0/0
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#attribute-flags 0x80000000
 ! mark interface as 0x80000000 which will exclude it from being used by TE251
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#int gi0/0/0/2
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#attribute-flags 0x80000000
 ! mark interface as 0x80000000 which will exclude it from being used by TE251
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#commit
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#
```

The end effect is the same:

```
RP/0/0/CPU0:SP2R2#sh rsvp session
Type Destination Add DPort  Proto/ExtTunID  PSBs  RSBs  Reqs
---- --------------- ----- --------------- ----- ----- -----
LSP4       12.39.5.5    25       12.39.2.2     1     1     0
LSP4       12.39.5.5   251       12.39.2.2     2     2     0

RP/0/0/CPU0:SP2R2#sh rsvp session detail
SESSION: IPv4-LSP Addr: 12.39.5.5, TunID: 25, ExtID: 12.39.2.2
[...]
   Explicit Route:
     Strict, 12.39.12.1/32
     Strict, 12.39.15.5/32
     Strict, 12.39.5.5/32

SESSION: IPv4-LSP Addr: 12.39.5.5, TunID: 251, ExtID: 12.39.2.2
[...]
   Explicit Route:
     Strict, 12.39.24.4/32
     Strict, 12.39.34.3/32
     Strict, 12.39.35.5/32
     Strict, 12.39.5.5/32
```

Tunnel 251 avoids Gi0/0/0/0 and Gi0/0/0/2 as those interfaces are invisible from CSPF point of view because of non-matching affinity value and mask.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 7.3: Link affinity with IOS-XE

Let's replicate the same idea of affinity configuration to IOS-XE. We have SP2R5 that has two core interfaces - Gi2 and Gi3. First, lets setup a vanilla tunnel with affinity mapping for 0x0/0xf:

```
SP2R5-RR#conf t
SP2R5-RR(config)#interface tunnel 52
SP2R5-RR(config-if)#ip unnumbered loopback0
SP2R5-RR(config-if)#tunnel destination 12.39.2.2
SP2R5-RR(config-if)#tunnel mode mpls traffic-eng
SP2R5-RR(config-if)#tunnel mpls traffic-eng affinity 0x0 mask 0xffffffff
SP2R5-RR(config-if)#tunnel mpls traffic-eng path-option 10 dynamic
SP2R5-RR(config-if)#^Z

SP2R5-RR#sh mpls traffic-eng tunnels brief
[...]
P2P TUNNELS/LSPs:
TUNNEL NAME                      DESTINATION      UP IF     DOWN IF   STATE/PROT
SP2R5-RR_t52                     12.39.2.2        -         Gi3       up/up
SP2R2_t25                        12.39.5.5        Gi3       -         up/up
SP2R2_t251                       12.39.5.5        Gi2       -         up/up
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 2 (of 2) tails
```

As Gi3 is choosen by MPLS TE, let's "break" this interface by defininig some bogus affinity value on the interface:

```
SP2R5-RR#conf t
SP2R5-RR(config-if)#int gi3
SP2R5-RR(config-if)#mpls traffic-eng attribute-flags 0x12345678
```

The tunnel will be shifted to Gi2 as Gi3 doesn't match any longer the required affinity value:

```
SP2R5-RR#sh mpls traffic-eng tunnels

P2P TUNNELS/LSPs:

Name: SP2R5-RR_t52                         (Tunnel52) Destination: 12.39.2.2
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, type dynamic (Basis for Setup, path weight 30)

[...]
  RSVP Signalling Info:
       Src 12.39.5.5, Dst 12.39.2.2, Tun_Id 52, Tun_Instance 2
    RSVP Path Info:
      My Address: 12.39.35.5
      Explicit Route: 12.39.35.3 12.39.13.1 12.39.12.2 12.39.2.2
      Record   Route:   NONE
      Tspec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
```

We can now also verify the selection process selected only interfaces with default value of affinity - 0x0:

```
SP2R5-RR#show mpls traffic-eng topology path tunnel 52
Query Parameters:
  Destination: 12.39.2.2
    Bandwidth: 0
   Priorities: 7 (setup), 7 (hold)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
     Affinity: 0x0 (value), 0xFFFFFFFF (mask)
Query Results:
  Min Bandwidth Along Path: 10000000 (kbps)
  Max Bandwidth Along Path: 10000000 (kbps)
  Node Hop Count: 3
  Hop   0: 12.39.35.5          : affinity 0x00000000, bandwidth 10000000 (kbps)
  Hop   1: 12.39.13.3          : affinity 0x00000000, bandwidth 10000000 (kbps)
  Hop   2: 12.39.12.1          : affinity 0x00000000, bandwidth 10000000 (kbps)
  Hop   3: 12.39.2.2
```

The same can be verified from any other router in IGP domain, like for example from SP2R2:

```
RP/0/0/CPU0:SP2R2#show mpls traffic-eng topology affinity 12.39.5.5

IGP Id: 0000.0000.0005.00, MPLS TE Id: 12.39.5.5 Router Node  (IS-IS AS1239 level-2)
  Link[0]:      Intf Address: 12.39.35.5, Nbr Intf Address: 12.39.35.3
      Attribute Flags: 0x0
      Ext Admin Group:
          Length: 256 bits
          Value : 0x::           ! this is Gi2 interface, going from R5 to R3
      Attribute Names:
  Link[1]:      Intf Address: 12.39.15.5, Nbr Intf Address: 12.39.15.1
      Attribute Flags: 0x12345678
      Ext Admin Group:
          Length: 256 bits
          Value : 0x::1234:5678  ! this is Gi3 interface, going from R5 to R1
      Attribute Names:
      Unnamed bit    : 3 4 5 6 9 10 12 14 18 20 21 25 28
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

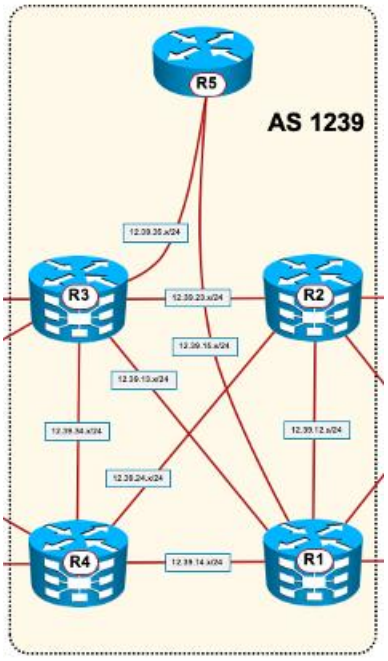
Task 7.4: Shared risk link groups (SRLG)

SRLG is another MPLS TE attribute (defined in RFC 4202, and then in RFC 4203 for OSPF and RFC 5307 for IS-IS), similar to affinity, but taken more as a preference rather than hard requirement. While AGs are hard (binary, either true or false), SRLG can be used when there should be clear preference to avoid fate-sharing, but not absolute or binary.



Before continuing, please remove all TE tunnels and affinity definitions in AS 1239 for them to not interfere with our exercises.

Let us consider the same topology within AS 1239:



Again, lets assume links from R2 to R1 and R3 are transported over one physical medium – link using DWDM system for example. This time our approach is a little bit different – we agree to build tunnel using DWDM link, bu the protection for that link should not use any other link from fate sharing group. First, we need to define the SRLGs themselves on the interfaces:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#srlg
RP/0/0/CPU0:SP2R2(config-srlg)#interface gigabitEthernet 0/0/0/2
RP/0/0/CPU0:SP2R2(config-srlg-if)#name DWDM1
RP/0/0/CPU0:SP2R2(config-srlg-if)#interface gigabitEthernet 0/0/0/0
RP/0/0/CPU0:SP2R2(config-srlg-if)#name DWDM1
RP/0/0/CPU0:SP2R2(config-srlg-if)#exit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP2R2(config-srlg)#name DWDM1 value 666
RP/0/0/CPU0:SP2R2(config-srlg-if)#commit
```

The definition should be visible on R2 after configuration:

```
RP/0/0/CPU0:SP2R2#show mpls traffic-eng topology srlg
My_System_id: 0000.0000.0002.00 (IS-IS AS1239 level-2)
My_BC_Model_Type: RDM

Signalling error holddown: 10 sec Global Link Generation 33319
                     SRLG Name      Admin      Interface    IGP Area  ID
                       (value)     Weight        Address
_____________________ _____________ _____ _____________ _____________
                    DWDM1(666)           1     12.39.12.2 IS-IS AS1239 level 2
                    DWDM1(666)           1     12.39.23.2 IS-IS AS1239 level 2

      SRLG        Admin Weight

_____________      _____________
        666                  1
```

However, as SRLG uses IGP to distribute information, the same information will be visible on SP2R4:

```
RP/0/0/CPU0:SP2R4#show mpls traffic-eng topology srlg
My_System_id: 0000.0000.0004.00 (IS-IS AS1239 level-2)
My_BC_Model_Type: RDM

Signalling error holddown: 10 sec Global Link Generation 33327
      SRLG   Admin Weight Interface Addr   TE Router ID IGP Area  ID
_____________ _____________ _____________ _____________ _____________
        666              1     12.39.12.2      12.39.2.2 IS-IS AS1239 level 2
        666              1     12.39.23.2      12.39.2.2 IS-IS AS1239 level 2
```

…as well as on SP2R5:

```
SP2R5-RR#show mpls traffic-eng topology srlg
(isis  level-2)
  SRLG:       Intf Addr:      MPLS TE Id:
  666         12.39.12.2      12.39.2.2
              12.39.23.2      12.39.2.2
```

Now we will use MPLS TE interface-level auto tunnel backup feature, to establish backup tunnels. Those backup tunnels should be taking into consideration SRLGs, and that requires explicit configuration.

First, we need to define how auto tunnels will be addressed, by enabling loopback 0 as source of IP addressing:

```
RP/0/0/CPU0:SP2R2(config)#ipv4 unnumbered mpls traffic-eng loopback 0
```

Now, we'll configure the auto-tunnel backup feature on interfaces, along with template to choose tunnel IDs from range – this is not strictly required, but makes potential troubleshooting easier:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#mpls traffic-eng
RP/0/0/CPU0:SP2R2(config-mpls-te)#interface gigabitEthernet 0/0/0/2
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#auto-tunnel backup
RP/0/0/CPU0:SP2R2(config-mpls-te-if-auto-backup)#exclude srlg
RP/0/0/CPU0:SP2R2(config-mpls-te-if-auto-backup)#int gi0/0/0/0
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#auto-tunnel backup
RP/0/0/CPU0:SP2R2(config-mpls-te-if-auto-backup)#exclude srlg
RP/0/0/CPU0:SP2R2(config-mpls-te-if-auto-backup)#int gi0/0/0/3
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
! note this interface is enabled for auto-tunnel backup but not limited by SRLG
RP/0/0/CPU0:SP2R2(config-mpls-te-if-auto-backup)#auto-tunnel backup
RP/0/0/CPU0:SP2R2(config-te-auto-bk)#tunnel-id min 3000 max 3999
RP/0/0/CPU0:SP2R2(config-te-auto-bk)#commit
```

The only thing left is tunnel itself – we'll ask for protection:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#interface tunnel-te 251
RP/0/0/CPU0:SP2R2(config-if)#ipv4 unnumbered Loopback0
RP/0/0/CPU0:SP2R2(config-if)#signalled-bandwidth 50000
RP/0/0/CPU0:SP2R2(config-if)#autoroute announce
RP/0/0/CPU0:SP2R2(config-if-tunte-aa)#destination 12.39.5.5
RP/0/0/CPU0:SP2R2(config-if)#fast-reroute
RP/0/0/CPU0:SP2R2(config-if)#path-option 10 dynamic
RP/0/0/CPU0:SP2R2(config-if)#commit
```

And the tunnel should be up with auto-tunnels set up automatically:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels

Name: tunnel-te251  Destination: 12.39.5.5  Ifhandle:0xf0
  Signalled-Name: SP2R2_t251
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type dynamic  (Basis for Setup, path weight 20)
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 50000 kbps  CT0
    Creation Time: Mon Jan 01 08:17:55 2018 (01:08:07 ago)
  Config Parameters:
    Bandwidth:    50000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Enabled, Protection Desired: Any
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  History:
    Tunnel has been up for: 00:29:19 (since Mon Jan 01 08:56:43 UTC 2018)
    Current LSP:
      Uptime: 00:29:19 (since Mon Jan 01 08:56:43 UTC 2018)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, identical to the [CURRENT] LSP
        Date/Time: Mon Jan 01 08:37:29 UTC 2018 [00:48:33 ago]
    Prior LSP:
      ID: 10 Path Option: 10
      Removal Trigger: reoptimization completed

  Path info (IS-IS AS1239 level-2):
  Node hop count: 2
  Hop0: 12.39.12.1
  Hop1: 12.39.15.5
  Hop2: 12.39.5.5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Name: tunnel-te3002  Destination: 12.39.5.5  Ifhandle:0x150 (auto-tunnel backup)
  Signalled-Name: autob_SP2R2_t3002_Gi0_0_0_2_12.39.1.1
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type explicit (autob_nnhop_srlg_te3002) (Basis for Setup, path
weight 30)
    path option 20,  type explicit (autob_nnhop_te3002) [disabled]
    SRLGs Excluded: DWDM1(666)
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Mon Jan 01 09:19:26 2018 (00:06:36 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  Auto Backup:
    Protected LSPs: 1
    Protected S2L Sharing Families: 0
    Protected S2L: 0
    Protected i/f: Gi0/0/0/2    Protected node: 12.39.1.1
    Attribute-set: Not configured
    Protection: NNHOP+SRLG (SRLG strict)
    Unused removal timeout: not running
  History:
    Tunnel has been up for: 00:06:36 (since Mon Jan 01 09:19:26 UTC 2018)
    Current LSP:
      Uptime: 00:06:36 (since Mon Jan 01 09:19:26 UTC 2018)

  Path info (IS-IS AS1239 level-2):
  Node hop count: 3
  Hop0: 12.39.24.4
  Hop1: 12.39.34.3
  Hop2: 12.39.35.5
  Hop3: 12.39.5.5

Name: tunnel-te3003  Destination: 12.39.1.1  Ifhandle:0x170 (auto-tunnel backup)
  Signalled-Name: autob_SP2R2_t3003_Gi0_0_0_2
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type explicit (autob_nhop_srlg_te3003) (Basis for Setup, path
weight 20)
    path option 20,  type explicit (autob_nhop_te3003) [disabled]
    SRLGs Excluded: DWDM1(666)
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Mon Jan 01 09:19:26 2018 (00:06:36 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  Auto Backup:
    Protected LSPs: 0
    Protected S2L Sharing Families: 0
    Protected S2L: 0
    Protected i/f: Gi0/0/0/2
    Attribute-set: Not configured
    Protection: NHOP+SRLG (SRLG strict)
    Unused removal timeout: 00:53:24
  History:
    Tunnel has been up for: 00:06:36 (since Mon Jan 01 09:19:26 UTC 2018)
    Current LSP:
      Uptime: 00:06:36 (since Mon Jan 01 09:19:26 UTC 2018)

  Path info (IS-IS AS1239 level-2):
  Node hop count: 2
  Hop0: 12.39.24.4
  Hop1: 12.39.14.1
  Hop2: 12.39.1.1
Displayed 3 (of 3) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 3 up, 0 down, 0 recovering, 0 recovered heads
```

Shorter version can be requested using:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels protection
SP2R2_t251 Tunnel Id: 251
  LSP Head, Admin: up, Oper: up
  Src: 12.39.2.2, Dest: 12.39.5.5, Instance: 12
  Fast Reroute Protection: Requested
    Outbound: FRR Ready
     Backup tunnel-te3002 to LSP nnhop
      tunnel-te3002: out i/f: GigabitEthernet0/0/0/3
     LSP signalling info:
      Original: out i/f: GigabitEthernet0/0/0/2, label:  123118, nhop: 12.39.12.1
      With FRR: out i/f: tunnel-te3002, label:  3
     LSP bw: 50000 kbps, Backup level: any-class unlimited, type: CT0

autob_SP2R2_t3002_Gi0_0_0_2_12.39.1.1 Tunnel Id: 3002
  LSP Head, Admin: up, Oper: up
  Src: 12.39.2.2, Dest: 12.39.5.5, Instance: 2
  Fast Reroute Protection: None


autob_SP2R2_t3003_Gi0_0_0_2 Tunnel Id: 3003
  LSP Head, Admin: up, Oper: up
  Src: 12.39.2.2, Dest: 12.39.1.1, Instance: 2
  Fast Reroute Protection: None
```

Note that both tunnels avoid Gi0/0/0/0 and Gi0/0/0/2 interfaces as defined in the SRLG. We can loosen the requirement, adding '**preferred**' to the statement on SRLG, and this is flexibility of SRLG contrasted with binary nature of attribute-groups – with SRLG as a last resort, we can use 'tainted' links.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Before doing that, let's check how SRLG and tunnel protection will behave when we'll shut down Gi0/0/0/3 – the only link not marked with SRLG attributes:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#int gi0/0/0/3
RP/0/0/CPU0:SP2R2(config-if)#shut
RP/0/0/CPU0:SP2R2(config-if)#commit
RP/0/0/CPU0:SP2R2(config-if)#
```

Now, let's check the status:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels

Name: tunnel-te251  Destination: 12.39.5.5  Ifhandle:0xf0
  Signalled-Name: SP2R2_t251
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
[…]
Name: tunnel-te3002  Destination: 12.39.5.5  Ifhandle:0x150 (auto-tunnel backup)
  Signalled-Name: autob_SP2R2_t3002_Gi0_0_0_2_12.39.1.1
  Status:
    Admin:    up Oper: down   Path: not valid   Signalling: Down

    path option 10,  type explicit (autob_nnhop_srlg_te3002)
    Last PCALC Error: Mon Jan 01 10:10:45 2018
      Info: No path to destination, 12.39.5.5 (exclude-srlg)
[…]

Name: tunnel-te3003  Destination: 12.39.1.1  Ifhandle:0x170 (auto-tunnel backup)
  Signalled-Name: autob_SP2R2_t3003_Gi0_0_0_2
  Status:
    Admin:    up Oper: down   Path: not valid   Signalling: Down

    path option 10,  type explicit (autob_nhop_srlg_te3003)
    Last PCALC Error: Mon Jan 01 10:10:45 2018
      Info: No path to destination, 12.39.1.1 (exclude-srlg)
[…]
```

Both backup tunnels are down, as Protection is defined as '**SRLG strict**'. When we reconfigure the policy, situation should change:

```
RP/0/0/CPU0:SP2R2#conf t
RP/0/0/CPU0:SP2R2(config)#mpls traffic-eng
RP/0/0/CPU0:SP2R2(config-mpls-te)#interface gigabitEthernet 0/0/0/0
RP/0/0/CPU0:SP2R2(config-mpls-te-if-auto-backup)#exclude srlg preferred
RP/0/0/CPU0:SP2R2(config-mpls-te-if-auto-backup)#interface gigabitEthernet 0/0/0/2
RP/0/0/CPU0:SP2R2(config-mpls-te-if)#auto-tunnel backup
RP/0/0/CPU0:SP2R2(config-mpls-te-if-auto-backup)#exclude srlg preferred
RP/0/0/CPU0:SP2R2(config-mpls-te-if-auto-backup)#commit
```

And the backup tunnels are up, showcasing in practice flexibility of SRLG and the main difference between binary affinity AGs and SRLGs:

```
RP/0/0/CPU0:SP2R2#sh mpls traffic-eng tunnels

Name: tunnel-te251  Destination: 12.39.5.5  Ifhandle:0xf0
  Signalled-Name: SP2R2_t251
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type dynamic  (Basis for Setup, path weight 20)
[…]
    Fast Reroute: Enabled, Protection Desired: Any
    Path Protection: Not Enabled
  Path info (IS-IS AS1239 level-2):
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Node hop count: 2
  Hop0: 12.39.12.1
  Hop1: 12.39.15.5
  Hop2: 12.39.5.5

Name: tunnel-te3002  Destination: 12.39.5.5  Ifhandle:0x150 (auto-tunnel backup)
  Signalled-Name: autob_SP2R2_t3002_Gi0_0_0_2_12.39.1.1
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 20,  type explicit (autob_nnhop_te3002) (Basis for Setup, path weight
20)
    path option 10,  type explicit (autob_nnhop_srlg_te3002)
 […]
    Protected LSPs: 1
    Protected S2L Sharing Families: 0
    Protected S2L: 0
    Protected i/f: Gi0/0/0/2    Protected node: 12.39.1.1
    Attribute-set: Not configured
    Protection: NNHOP (SRLG preferred)
    Unused removal timeout: not running
[…]
  Path info (IS-IS AS1239 level-2):
  Node hop count: 2
  Hop0: 12.39.23.3
  Hop1: 12.39.35.5
  Hop2: 12.39.5.5

Name: tunnel-te3003  Destination: 12.39.1.1  Ifhandle:0x170 (auto-tunnel backup)
  Signalled-Name: autob_SP2R2_t3003_Gi0_0_0_2
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 20,  type explicit (autob_nhop_te3003) (Basis for Setup, path weight 20)
    path option 10,  type explicit (autob_nhop_srlg_te3003)
[…]
    Protected LSPs: 0
    Protected S2L Sharing Families: 0
    Protected S2L: 0
    Protected i/f: Gi0/0/0/2
    Attribute-set: Not configured
    Protection: NHOP (SRLG preferred)
[…]
  Path info (IS-IS AS1239 level-2):
  Node hop count: 2
  Hop0: 12.39.23.3
  Hop1: 12.39.13.1
  Hop2: 12.39.1.1
Displayed 3 (of 3) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 3 up, 0 down, 0 recovering, 0 recovered heads
```

Remove the SRLGs and Tunnels before continuing with the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 8: MPLS TE auto mesh

Task 8.1: Configuring MPLS TE auto-tunnel mesh

You can scale up the MPLS TE deployment by using auto mesh functionality.

First step is to enable this feature on MPLS TE level:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#mpls traffic-eng
RP/0/0/CPU0:SP1R4(config-mpls-te)#auto-tunnel mesh
RP/0/0/CPU0:SP1R4(config-te-auto-mesh)#
```

Next, you can either enable all interfaces present in LDP to establish tunnels, or specify specific TE Ids. Let's go with the latter – we'll configure a list of our TE Ids in AS3356:

```
RP/0/0/CPU0:SP1R4#show mpls traffic-eng topology | include TE Id
IGP Id: 33.56.1.1, MPLS TE Id: 33.56.1.1 Router Node  (OSPF AS3356 area 0)
IGP Id: 33.56.2.2, MPLS TE Id: 33.56.2.2 Router Node  (OSPF AS3356 area 0)
IGP Id: 33.56.3.3, MPLS TE Id: 33.56.3.3 Router Node  (OSPF AS3356 area 0)
IGP Id: 33.56.4.4, MPLS TE Id: 33.56.4.4 Router Node  (OSPF AS3356 area 0)
IGP Id: 33.56.5.5, MPLS TE Id: 33.56.5.5 Router Node  (OSPF AS3356 area 0)
```

So the prefix-list will contain:

```
RP/0/0/CPU0:SP1R4(config)#ipv4 prefix-list MESH
RP/0/0/CPU0:SP1R4(config-ipv4_pfx)#permit 33.56.1.1/32
RP/0/0/CPU0:SP1R4(config-ipv4_pfx)#permit 33.56.2.2/32
RP/0/0/CPU0:SP1R4(config-ipv4_pfx)#permit 33.56.3.3/32
RP/0/0/CPU0:SP1R4(config-ipv4_pfx)#permit 33.56.4.4/32
RP/0/0/CPU0:SP1R4(config-ipv4_pfx)#permit 33.56.5.5/32
RP/0/0/CPU0:SP1R4(config-ipv4_pfx)#commit
RP/0/0/CPU0:SP1R4(config-ipv4_pfx)#
```

For the auto tunnels, we need to define both the source of tunnel and the numbering scheme:

```
RP/0/0/CPU0:SP1R2(config)#ipv4 unnumbered mpls traffic-eng Loopback 0
```

Next step is to configure the template Auto-Tunnel mesh feature will use to spawn new tunnels:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#mpls traffic-eng
RP/0/0/CPU0:SP1R4(config-mpls-te)#attribute-set auto-mesh AS3356-MESH
RP/0/0/CPU0:SP1R4(config-te-attribute-set)#signalled-bandwidth 1000
RP/0/0/CPU0:SP1R4(config-te-attribute-set)#autoroute announce
RP/0/0/CPU0:SP1R4(config-te-attribute-set)#commit
```

Define the group with those attributes and destination list:

```
RP/0/0/CPU0:SP1R4(config-mpls-te)#auto-tunnel mesh
RP/0/0/CPU0:SP1R4(config-te-auto-mesh)#group 100
RP/0/0/CPU0:SP1R4(config-te-auto-mesh-group)#destination-list MESH
RP/0/0/CPU0:SP1R4(config-te-auto-mesh-group)#attribute-set AS3356-MESH
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Decide about numbering scheme for the tunnels to be spawned:

```
RP/0/0/CPU0:SP1R2(config-te-auto-mesh)#tunnel-id min 5000 max 6000
```

And finally, enable the group we just configured for auto-tunnel mesh:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#mpls ldp
RP/0/0/CPU0:SP1R4(config-ldp)#address-family ipv4
RP/0/0/CPU0:SP1R4(config-ldp-af)#traffic-eng auto-tunnel mesh
RP/0/0/CPU0:SP1R4(config-ldp-af-te-mesh)#group 100
RP/0/0/CPU0:SP1R4(config-ldp-af-te-mesh)#commit
RP/0/0/CPU0:SP1R4(config-ldp-af-te-mesh)#
```

After repeating this configuration on SP1, SP2, SP3, let's focus on SP5 which requires similar, but slightly different configuration thanks to its IOS-XE origin:

```
SP1-R5-RR#conf t
SP1-R5-RR(config)#mpls traffic-eng auto-tunnel mesh tunnel-num min 5000 max 6000
SP1-R5-RR(config)#interface auto-template 100
SP1-R5-RR(config-if)#ip unnumbered loopback 0
SP1-R5-RR(config-if)#tunnel destination access-list 1
SP1-R5-RR(config-if)#tunnel mode mpls traffic-eng
SP1-R5-RR(config-if)#tunnel mpls traffic-eng path-option 10 dynamic
SP1-R5-RR(config-if)#tunnel mpls traffic-eng autoroute announce
SP1-R5-RR(config)#ip access-list standard 1
SP1-R5-RR(config-std-nacl)#permit 33.56.1.1
SP1-R5-RR(config-std-nacl)#permit 33.56.2.2
SP1-R5-RR(config-std-nacl)#permit 33.56.4.4
SP1-R5-RR(config-std-nacl)#permit 33.56.5.5
SP1-R5-RR(config-std-nacl)#^Z
```

We can check the config and situation on both IOS-XE node:

```
SP1-R5-RR#sh mpls traffic-eng tunnels brief
Signalling Summary:
    LSP Tunnels Process:            running
    Passive LSP Listener:           running
    RSVP Process:                   running
    Forwarding:                     enabled
    auto-tunnel:
        backup Disabled (0 ), id-range:65436-65535
        onehop Disabled (0 ), id-range:65336-65435
        mesh   Enabled  (4 ), id-range:5000-6000
        p2p    Disabled (0), id-range:62336-64335

    Periodic reoptimization:        every 3600 seconds, next in 2427 seconds
    Periodic FRR Promotion:         Not Running
    Periodic auto-bw collection:    every 300 seconds, next in 27 seconds
    SR tunnel max label push:       13 labels

P2P TUNNELS/LSPs:
TUNNEL NAME                      DESTINATION      UP IF      DOWN IF    STATE/PROT
SP1-R5-RR_t5000                  33.56.1.1        -          Gi3        up/up
SP1-R5-RR_t5001                  33.56.2.2        -          Gi2        up/up
SP1-R5-RR_t5002                  33.56.3.3        -          Gi2        up/up
SP1-R5-RR_t5003                  33.56.4.4        -          Gi2        up/up
autom_SP1R1_t5003_mg100          33.56.5.5        Gi3        -          up/up
autom_SP1R2_t5003_mg100          33.56.5.5        Gi2        -          up/up
autom_SP1R3_t5003_mg100          33.56.5.5        Gi2        -          up/up
autom_SP1R4_t5003_mg100          33.56.5.5        Gi3        -          up/up
Displayed 4 (of 4) heads, 0 (of 0) midpoints, 4 (of 4) tails

P2MP TUNNELS:
Displayed 0 (of 0) P2MP heads
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
P2MP SUB-LSPS:
Displayed 0 P2MP sub-LSPs:
          0 (of 0) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

Let's check the resulting MPLS TE mesh:

```
SP1-R5-RR#sh mpls traffic-eng tunnels summary
Signalling Summary:
    LSP Tunnels Process:            running
    Passive LSP Listener:           running
    RSVP Process:                   running
    Forwarding:                     enabled
    auto-tunnel:
        backup Disabled (0 ), id-range:65436-65535
        onehop Disabled (0 ), id-range:65336-65435
        mesh   Enabled  (4 ), id-range:5000-6000
        p2p    Disabled (0), id-range:62336-64335

    Periodic reoptimization:        every 3600 seconds, next in 2283 seconds
    Periodic FRR Promotion:         Not Running
    Periodic auto-bw collection:    every 300 seconds, next in 183 seconds
    SR tunnel max label push:       13 labels
    P2P:
      Head: 4 interfaces,   4 active signalling attempts, 4 established
            4 activations,  0 deactivations
            0 SSO recovery attempts, 0 SSO recovered
      Midpoints: 0, Tails: 4

    P2MP:
      Head: 0 interfaces,   0 active signalling attempts, 0 established
            0 sub-LSP activations,  0 sub-LSP deactivations
            0 LSP successful activations,  0 LSP deactivations
            0 SSO recovery attempts, LSP recovered: 0 full, 0 partial, 0 fail
      Midpoints: 0, Tails: 0

Bidirectional Tunnel Summary:
    Tunnel Head: 0 total, 0 connected, 0 associated, 0 co-routed
    LSPs Head:   0 established, 0 proceeding, 0 associated, 0 standby
    LSPs Mid:    0 established, 0 proceeding, 0 associated, 0 standby
    LSPs Tail:   0 established, 0 proceeding, 0 associated, 0 standby

SP1-R5-RR#sh ip route ospf | i /32
O        33.56.1.1/32 [110/2] via 33.56.1.1, 00:06:07, Tunnel5000
O        33.56.2.2/32 [110/3] via 33.56.2.2, 00:06:07, Tunnel5001
O        33.56.3.3/32 [110/2] via 33.56.3.3, 00:06:07, Tunnel5002
O        33.56.4.4/32 [110/3] via 33.56.4.4, 00:06:07, Tunnel5003

SP1-R5-RR#sh ip rsvp reservation
To              From            Pro DPort Sport Next Hop        I/F      Fi Serv BPS
33.56.1.1       33.56.5.5       0   5000  3205  33.56.15.1      Gi3      SE LOAD 0
33.56.2.2       33.56.5.5       0   5001  6536  33.56.35.3      Gi2      SE LOAD 0
33.56.3.3       33.56.5.5       0   5002  2523  33.56.35.3      Gi2      SE LOAD 0
33.56.4.4       33.56.5.5       0   5003  5965  33.56.35.3      Gi2      SE LOAD 0
33.56.5.5       33.56.1.1       0   5003  3     none            none     SE LOAD 1M
33.56.5.5       33.56.2.2       0   5003  3     none            none     SE LOAD 1M
33.56.5.5       33.56.3.3       0   5003  2     none            none     SE LOAD 1M
33.56.5.5       33.56.4.4       0   5003  3     none            none     SE LOAD 1M
```

…and on IOS XR node:

```
RP/0/0/CPU0:SP1R4#sh mpls traffic-eng tunnels brief

                      TUNNEL NAME          DESTINATION      STATUS  STATE
                    +tunnel-te5000           33.56.1.1          up  up
                    +tunnel-te5001           33.56.2.2          up  up
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
                  +tunnel-te5002           33.56.3.3          up  up
                  +tunnel-te5003           33.56.5.5          up  up
         autom_SP1R1_t5002_mg100           33.56.4.4          up  up
         autom_SP1R1_t5002_mg100           33.56.4.4          up  up
         autom_SP1R2_t5002_mg100           33.56.4.4          up  up
         autom_SP1R2_t5002_mg100           33.56.4.4          up  up
         autom_SP1R3_t5002_mg100           33.56.4.4          up  up

RP/0/0/CPU0:SP1R4#sh mpls forwarding | i 33.56. | i /32
104000 Exp-Null-v4 33.56.2.2/32       tt5001       33.56.2.2       892
104001 Exp-Null-v4 33.56.1.1/32       tt5000       33.56.1.1       106
104002 Exp-Null-v4 33.56.3.3/32       tt5002       33.56.3.3       150
104003 Pop         33.56.5.5/32       tt5003       33.56.5.5       0
104032 103008      3:3:172.16.3.31/32              33.56.3.3       0
104066 103037      12.39.2.2/32       tt5002       33.56.3.3       0

RP/0/0/CPU0:SP1R4#sh rsvp reservation
Destination Add DPort      Source Add SPort Pro   Input IF Sty Serv   Rate Burst
--------------- ----- --------------- ----- --- ---------- --- ---- ------ -----
      33.56.1.1  5000       33.56.4.4     3   0  Gi0/0/0/0  SE LOAD  1000K    1K
      33.56.2.2  5001       33.56.4.4     3   0  Gi0/0/0/5  SE LOAD  1000K    1K
      33.56.3.3  5002       33.56.4.4     3   0  Gi0/0/0/2  SE LOAD  1000K    1K
      33.56.4.4  5002       33.56.1.1     3   0         No  SE LOAD  1000K    1K
      33.56.4.4  5002       33.56.2.2     3   0         No  SE LOAD  1000K    1K
      33.56.4.4  5002       33.56.3.3     2   0         No  SE LOAD  1000K    1K
      33.56.4.4  5003       33.56.5.5  5965   0         No  SE LOAD     0     1K
      33.56.5.5  5003       33.56.4.4     3   0  Gi0/0/0/0  SE LOAD  1000K    1K
```

And for the tunnels (note, only one tunnel established – SP1R2 to SP1R1):

```
RP/0/0/CPU0:SP1R2#sh mpls traffic-eng auto-tunnel mesh
Auto-tunnel Mesh Global Configuration:
  Unused removal timeout: 1h 0m 0s
  Configured tunnel number range: 5000-6000

Auto-tunnel Mesh Groups Summary:
  Mesh Groups count: 1
  Mesh Groups Destinations count: 1
  Mesh Groups Tunnels count:
    1 created, 1 up, 0 down, 0 FRR enabled

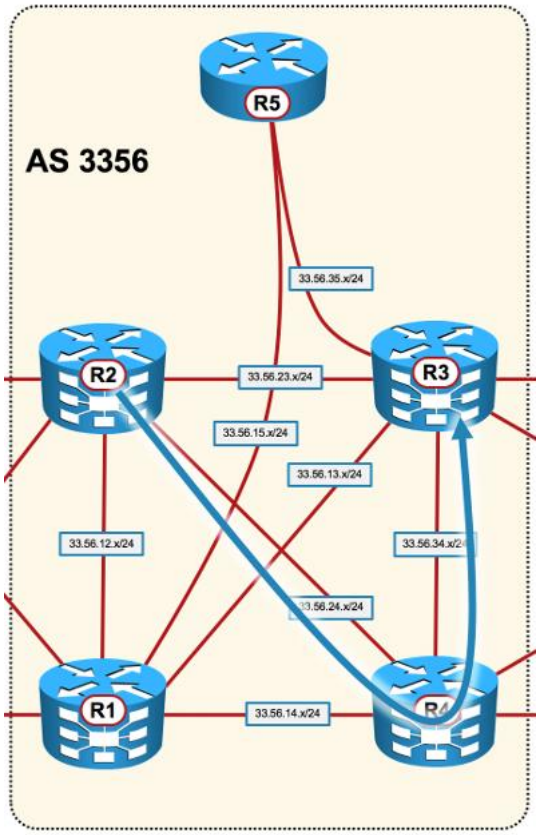
Mesh Group: 100 (1 Destinations)
  Status: Enabled
  Attribute-set: AS3356-MESH
  Destination-list: MESH
  Recreate timer: Not running
       Destination    Tunnel ID    State  Unused timer
  ----------------  -----------  -------  ------------
         33.56.1.1         5000       up   Not running
  Displayed 1 tunnels, 1 up, 0 down, 0 FRR enabled

Auto-mesh Cumulative Counters:
  Last cleared: Tue Jan  1 10:16:49 2017 (00:59:43 ago)
                     Total
  Created:               1
  Connected:             1
  Removed (unused):      0
  Removed (in use):      0
  Range exceeded:        0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 9: Configuring PE-P MPLS TE tunnels

We'll modify a tunnel going to 12.39.2.2 to end on R3:



Let's verify the proper path traffic takes within VPN service:

```
RP/0/0/CPU0:SP1R2#traceroute vrf cust1 192.168.7.71

Type escape sequence to abort.
Tracing the route to 192.168.7.71

 1  33.56.24.4 [MPLS: Labels 104064/123220 Exp 0] 19 msec  9 msec  9 msec
 2  33.56.34.3 [MPLS: Labels 103038/123220 Exp 0] 9 msec  9 msec  9 msec
 3  22.22.33.254 [MPLS: Labels 123319/123220 Exp 0] 9 msec  9 msec  9 msec
 4  12.39.23.2 [MPLS: Label 123220 Exp 0] 9 msec  9 msec  9 msec
 5  20.40.27.7 9 msec  *  9 msec
```

Now, we'll shut down R2-R3 link:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#int gi0/0/0/0
RP/0/0/CPU0:SP1R2(config-if)#shut
RP/0/0/CPU0:SP1R2(config-if)#commit
```

We have to modify tunnel destination and the path tunnel should take:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#interface tunnel-te24332
RP/0/0/CPU0:SP1R2(config-if)#destination 33.56.3.3
RP/0/0/CPU0:SP1R2(config-if)#exit
RP/0/0/CPU0:SP1R2(config)#explicit-path name InterAS-TE
RP/0/0/CPU0:SP1R2(config-expl-path)#no index 30
RP/0/0/CPU0:SP1R2(config-expl-path)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Verify that the tunnel is back up again:

```
RP/0/0/CPU0:SP1R2#sh mpls traffic-eng tunnels
Name: tunnel-te24332  Destination: 33.56.3.3  Ifhandle:0x110
  Signalled-Name: SP1R2_t24332
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type explicit InterAS-TE (Basis for Setup, path weight 2)
    Last Signalled Error : Thu May 11 21:30:58 2017
      Info: [2] PathErr(24,5)-(routing, no route to dest) at 33.56.34.3
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 5000 kbps  CT0
    Creation Time: Thu May 11 21:28:32 2017 (1d00h ago)
  Config Parameters:
    Bandwidth:     5000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  History:
    Tunnel has been up for: 00:00:03 (since Fri May 12 22:22:09 UTC 2017)
    Current LSP:
      Uptime: 00:00:03 (since Fri May 12 22:22:09 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Fri May 12 22:22:08 UTC 2017 [00:00:04 ago]
    Prior LSP:
      ID: 3 Path Option: 10
      Removal Trigger: path tear

  Path info (OSPF AS3356 area 0):
  Node hop count: 2
  Hop0: 33.56.24.4
  Hop1: 33.56.34.3
  Hop2: 33.56.3.3
```

But right now we have a problem with the VPN service:

```
RP/0/0/CPU0:SP1R2#traceroute 192.168.7.71 vrf cust1

Type escape sequence to abort.
Tracing the route to 192.168.7.71

 1    *  *
RP/0/0/CPU0:SP1R2#ping 192.168.7.71 vrf cust1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.7.71, timeout is 2 seconds:
...
Success rate is 0 percent (0/3)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

…and tracing within global routing table hints us, that we're missing label on R4->R3 link, which makes R3 try to interpret VPN label as transport label (and failing):

```
RP/0/0/CPU0:SP1R2#traceroute 12.39.2.2 source 33.56.2.2

Type escape sequence to abort.
Tracing the route to 12.39.2.2

 1  33.56.24.4 [MPLS: Label 104023 Exp 0] 9 msec  0 msec  0 msec
 2  33.56.34.3 0 msec  0 msec  0 msec ! look Ma'am! no labels! ☺
 3  22.22.33.254 [MPLS: Label 123302 Exp 0] 9 msec  9 msec  9 msec
 4  12.39.23.2 9 msec  *  9 msec
```

On SP1R3 enable accepting targeted hellos:

```
RP/0/0/CPU0:SP1R3#conf t
RP/0/0/CPU0:SP1R3(config)#mpls ldp
RP/0/0/CPU0:SP1R3(config-ldp)#address-family ipv4
RP/0/0/CPU0:SP1R3(config-ldp-af)#discovery targeted-hello accept
RP/0/0/CPU0:SP1R3(config-ldp-af)#commit
```

Back on SP1R2 enable LDP on tunnel:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#mpls ldp
RP/0/0/CPU0:SP1R2(config-ldp)#interface tunnel-te24332
RP/0/0/CPU0:SP1R2(config-ldp-if)#commit
```

Verify targeted hello is working:

```
RP/0/0/CPU0:SP1R2#sh mpls ldp discovery

Local LDP Identifier: 33.56.2.2:0
Discovery Sources:
  Interfaces:
    GigabitEthernet0/0/0/2 : xmit/recv
      VRF: 'default' (0x60000000)
      LDP Id: 33.56.1.1:0, Transport address: 33.56.1.1
          Hold time: 15 sec (local:15 sec, peer:15 sec)
          Established: May 11 21:28:36.803 (1d01h ago)

    GigabitEthernet0/0/0/4 : xmit/recv
      VRF: 'default' (0x60000000)
      LDP Id: 33.56.4.4:0, Transport address: 33.56.4.4
          Hold time: 15 sec (local:15 sec, peer:15 sec)
          Established: May 11 21:28:56.501 (1d01h ago)

    tunnel-te24332 : xmit
      Target: 33.56.3.3

  Targeted Hellos:
    33.56.2.2 -> 33.56.3.3 (active), xmit/recv
      LDP Id: 33.56.3.3:0
          Hold time: 90 sec (local:90 sec, peer:90 sec)
          Established: May 12 22:57:05.350 (00:01:56 ago)
```

And tunnel interface is up from LDP perspective:

```
RP/0/0/CPU0:SP1R2#sh mpls ldp interface tunnel-te24332
Interface tunnel-te24332 (0x110)
   VRF: 'default' (0x60000000)
   Enabled via config: LDP interface
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Let's test the traffic within VPN:

```
RP/0/0/CPU0:SP1R2#traceroute vrf cust1 192.168.7.71

Type escape sequence to abort.
Tracing the route to 192.168.7.71

 1  33.56.24.4 [MPLS: Labels 104023/103037/123220 Exp 0] 9 msec  9 msec  9 msec
 2  33.56.34.3 [MPLS: Labels 103037/123220 Exp 0] 9 msec  9 msec  9 msec
 3  22.22.33.254 [MPLS: Labels 123302/123220 Exp 0] 9 msec  9 msec  9 msec
 4  12.39.23.2 [MPLS: Labels 0/123220 Exp 0] 9 msec  9 msec  9 msec
 5  20.40.27.7 19 msec  *  9 msec
```

Notice LDP signalled label:

```
RP/0/0/CPU0:SP1R2#sh mpls ldp forwarding 12.39.2.2/32

Codes:
  - = GR label recovering, (!) = LFA FRR pure backup path
  {} = Label stack with multi-line output for a routing path
  G = GR, S = Stale, R = Remote LFA FRR backup

Prefix          Label   Label(s)       Outgoing     Next Hop            Flags
                In      Out            Interface                        G S R
--------------- ------- -------------- ------------ ------------------- -----
12.39.2.2/32    102029  103037         tt24332      point2point
```

As well the RSVP one (that can be found in the upper traceroute example):

```
RP/0/0/CPU0:SP1R2#sh rsvp reservation detail
RESV: IPv4-LSP Session addr: 33.56.3.3. TunID: 24332. LSPId: 5.
 Source addr: 33.56.2.2. ExtID: 33.56.2.2.
 Input adjusted interface: Gi0/0/0/4. Input physical interface: Gi0/0/0/4.
 Next hop: 33.56.24.4 (lih: 0x8).
 Style: Shared-Explicit. Service: Controlled-Load.
 Rate: 5000000 bits/sec. Burst: 1K bytes. Peak: 5M bits/sec.
 MTU min: 40, max: 1500 bytes.
 Flags: None.
 State expires in 350.031 sec.
 Policy:  Accepted. Policy source(s): Default.
 Header info: RSVP TTL=253. IP TTL=253. Flags: 0x1. TOS=0xc0.
 Resource:
  Labels: Outgoing downstream: 104023.
```

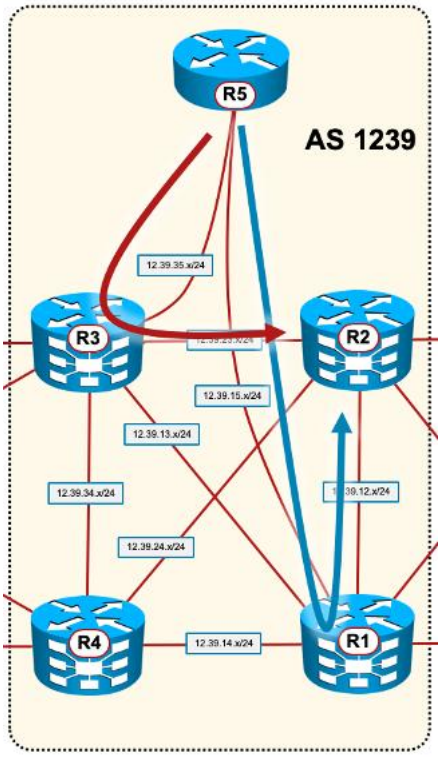

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 10: MPLS TE load balancing

Task 10.1: Configuring MPLS TE load balancing with IOS XE

If we want to do load balancing with IOS XE we need to setup at least two tunnels going to the same destination, but make sure they'll follow different physical paths (otherwise the load balancing doesn't make any sense). You can use different means to achieve different paths (like affinity or SRLG), but we'll use simplest option – explicit path definition.

So, starting from SP2R5 we'll configure two tunnels to SP2R2 – one via SP2R1 (blue) and one via SP2R3 (red):




```
SP2R5-RR#conf t
SP2R5-RR(config)#interface tunnel512
SP2R5-RR(config-if)#ip unnumbered loopback 0
SP2R5-RR(config-if)#tunnel mode mpls traffic-eng
SP2R5-RR(config-if)#tunnel mpls traffic-eng autoroute announce
SP2R5-RR(config-if)#tunnel destination 12.39.2.2
SP2R5-RR(config-if)#tunnel mpls traffic-eng path-option 10 explicit name TE512
SP2R5-RR(config-if)#exit
SP2R5-RR(config)#interface tunnel532
SP2R5-RR(config-if)#ip unnumbered loopback 0
SP2R5-RR(config-if)#tunnel destination 12.39.2.2
SP2R5-RR(config-if)#tunnel mode mpls traffic-eng
SP2R5-RR(config-if)#tunnel mpls traffic-eng autoroute announce
SP2R5-RR(config-if)#tunnel mpls traffic-eng path-option 10 explicit name TE532
SP2R5-RR(config)#ip explicit-path name TE512
SP2R5-RR(cfg-ip-expl-path)#next-address 12.39.1.1
Explicit Path name TE512:
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    1: next-address 12.39.1.1
SP2R5-RR(cfg-ip-expl-path)#next-address 12.39.2.2
Explicit Path name TE512:
    1: next-address 12.39.1.1
    2: next-address 12.39.2.2
SP2R5-RR(cfg-ip-expl-path)#ip explicit-path name TE532
SP2R5-RR(cfg-ip-expl-path)#next-address 12.39.3.3
Explicit Path name TE532:
    1: next-address 12.39.3.3
SP2R5-RR(cfg-ip-expl-path)#next-address 12.39.2.2
Explicit Path name TE532:
    1: next-address 12.39.3.3
    2: next-address 12.39.2.2
```

After such configuration, both tunnels should be up and the routing table shoud have load balancing entry for 12.39.2.2/32 using both tunnels 1:1:

```
SP2R5-RR#sh ip route 12.39.2.2
Routing entry for 12.39.2.2/32
  Known via "isis", distance 115, metric 30, type level-2
  Redistributing via isis AS1239
  Last update from 12.39.2.2 on Tunnel532, 00:00:38 ago
  Routing Descriptor Blocks:
    12.39.2.2, from 12.39.2.2, 00:00:38 ago, via Tunnel532
      Route metric is 30, traffic share count is 1
  * 12.39.2.2, from 12.39.2.2, 00:00:38 ago, via Tunnel512
      Route metric is 30, traffic share count is 1
```

The 1:1 ratio can be verified with CEF:

```
SP2R5-RR#sh ip cef 12.39.2.2 internal
12.39.2.2/32, epoch 2, RIB[I], refcnt 6, per-destination sharing
  sources: RIB, LTE
  feature space:
    IPRM: 0x00028000
    Broker: linked, distributed at 4th priority
    LFD: 12.39.2.2/32 1 local label
    dflt local label info: global/24 [0x3]
        contains path extension list
        dflt disposition chain 0x7FBDFCC5F6C0
          loadinfo 80007FBDFCC5F6C0, per-session, 2 choices, flags 0009, 5 locks
        dflt label switch chain 0x7FBDFCC5F980
          loadinfo 80007FBDFCC5F980, per-session, 2 choices, flags 0011, 5 locks
  ifnums:
    Tunnel512(14)
    Tunnel532(15)
  path list 7FBE643EAD10, 3 locks, per-destination, flags 0x49 [shble, rif, hwcn]
    path 7FBE643EDB18, share 1/1, type attached nexthop, for IPv4
      MPLS short path extensions: [none] MOI flags = 0x1 label implicit-null
      nexthop 12.39.2.2 Tunnel512, IP midchain out of Tunnel512 7FBE5C6B8B18
    path 7FBE643EDBC8, share 1/1, type attached nexthop, for IPv4
      MPLS short path extensions: [none] MOI flags = 0x1 label implicit-null
      nexthop 12.39.2.2 Tunnel532, IP midchain out of Tunnel532 7FBE5C6B8738
  output chain:
    loadinfo 80007FBDFCC5FF00, per-session, 2 choices, flags 0003, 5 locks
      flags [Per-session, for-rx-IPv4]
      16 hash buckets
        < 0 > IP midchain out of Tunnel512 7FBE5C6B8B18
              label 123116
              TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
        < 1 > IP midchain out of Tunnel532 7FBE5C6B8738
              label 123310
              TAG adj out of GigabitEthernet2, addr 12.39.35.3 7FBE5C6B94C8
        < 2 > IP midchain out of Tunnel512 7FBE5C6B8B18
              label 123116
              TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
        < 3 > IP midchain out of Tunnel532 7FBE5C6B8738
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
               label 123310
               TAG adj out of GigabitEthernet2, addr 12.39.35.3 7FBE5C6B94C8
       < 4 > IP midchain out of Tunnel512 7FBE5C6B8B18
               label 123116
               TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
       < 5 > IP midchain out of Tunnel532 7FBE5C6B8738
               label 123310
               TAG adj out of GigabitEthernet2, addr 12.39.35.3 7FBE5C6B94C8
       < 6 > IP midchain out of Tunnel512 7FBE5C6B8B18
               label 123116
               TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
       < 7 > IP midchain out of Tunnel532 7FBE5C6B8738
               label 123310
               TAG adj out of GigabitEthernet2, addr 12.39.35.3 7FBE5C6B94C8
       < 8 > IP midchain out of Tunnel512 7FBE5C6B8B18
               label 123116
               TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
       < 9 > IP midchain out of Tunnel532 7FBE5C6B8738
               label 123310
               TAG adj out of GigabitEthernet2, addr 12.39.35.3 7FBE5C6B94C8
       <10 > IP midchain out of Tunnel512 7FBE5C6B8B18
               label 123116
               TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
       <11 > IP midchain out of Tunnel532 7FBE5C6B8738
               label 123310
               TAG adj out of GigabitEthernet2, addr 12.39.35.3 7FBE5C6B94C8
       <12 > IP midchain out of Tunnel512 7FBE5C6B8B18
               label 123116
               TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
       <13 > IP midchain out of Tunnel532 7FBE5C6B8738
               label 123310
               TAG adj out of GigabitEthernet2, addr 12.39.35.3 7FBE5C6B94C8
       <14 > IP midchain out of Tunnel512 7FBE5C6B8B18
               label 123116
               TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
       <15 > IP midchain out of Tunnel532 7FBE5C6B8738
               label 123310
               TAG adj out of GigabitEthernet2, addr 12.39.35.3 7FBE5C6B94C8
     Subblocks:
       None
```

If we want to change the load-sharing weighting, this needs to be adjusted at tunnel level. For example, lets assume Tu512 can transport 5 times more traffic than Tu532:

```
SP2R5-RR#conf t
SP2R5-RR(config)#interface tunnel 512
SP2R5-RR(config-if)#tunnel mpls traffic-eng load-share 5
SP2R5-RR(config)#interface tunnel 532
SP2R5-RR(config-if)#tunnel mpls traffic-eng load-share 1
```

Now the split should look 5:1:

```
SP2R5-RR#sh ip route 12.39.2.2
Routing entry for 12.39.2.2/32
  Known via "isis", distance 115, metric 30, type level-2
  Redistributing via isis AS1239
  Last update from 12.39.2.2 on Tunnel532, 00:00:11 ago
  Routing Descriptor Blocks:
    12.39.2.2, from 12.39.2.2, 00:00:11 ago, via Tunnel532
      Route metric is 30, traffic share count is 1
  * 12.39.2.2, from 12.39.2.2, 00:00:11 ago, via Tunnel512
      Route metric is 30, traffic share count is 5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

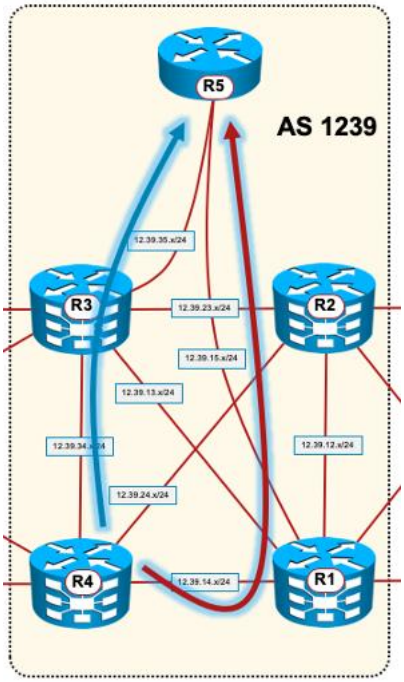
And this again can be verified with CEF:

```
SP2R5-RR#sh ip cef 12.39.2.2 internal
12.39.2.2/32, epoch 2, RIB[I], refcnt 6, per-destination sharing
  sources: RIB, LTE
  feature space:
    IPRM: 0x00028000
    Broker: linked, distributed at 4th priority
    LFD: 12.39.2.2/32 1 local label
    dflt local label info: global/24 [0x3]
        contains path extension list
        dflt disposition chain 0x7FBDFCC5F820
          loadinfo 80007FBDFCC5F820, per-session, 2 choices, flags 0009, 5 locks
        dflt label switch chain 0x7FBDFCC5F2A0
          loadinfo 80007FBDFCC5F2A0, per-session, 2 choices, flags 0011, 5 locks
  ifnums:
    Tunnel512(14)
    Tunnel532(15)
  path list 7FBE643EADB0, 3 locks, per-destination, flags 0x49 [shble, rif, hwcn]
    path 7FBE643EDDD8, share 1/1, type attached nexthop, for IPv4
      MPLS short path extensions: [none] MOI flags = 0x1 label implicit-null
      nexthop 12.39.2.2 Tunnel532, IP midchain out of Tunnel532 7FBE5C6B8738
    path 7FBE643EDD28, share 5/5, type attached nexthop, for IPv4
      MPLS short path extensions: [none] MOI flags = 0x1 label implicit-null
      nexthop 12.39.2.2 Tunnel512, IP midchain out of Tunnel512 7FBE5C6B8B18
  output chain:
    loadinfo 80007FBDFCC5FAE0, per-session, 2 choices, flags 0003, 5 locks
      flags [Per-session, for-rx-IPv4]
      12 hash buckets
        < 0 > IP midchain out of Tunnel532 7FBE5C6B8738
              label 123310
              TAG adj out of GigabitEthernet2, addr 12.39.35.3 7FBE5C6B94C8
        < 1 > IP midchain out of Tunnel512 7FBE5C6B8B18
              label 123116
              TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
        < 2 > IP midchain out of Tunnel532 7FBE5C6B8738
              label 123310
              TAG adj out of GigabitEthernet2, addr 12.39.35.3 7FBE5C6B94C8
        < 3 > IP midchain out of Tunnel512 7FBE5C6B8B18
              label 123116
              TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
        < 4 > IP midchain out of Tunnel512 7FBE5C6B8B18
              label 123116
              TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
        < 5 > IP midchain out of Tunnel512 7FBE5C6B8B18
              label 123116
              TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
        < 6 > IP midchain out of Tunnel512 7FBE5C6B8B18
              label 123116
              TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
        < 7 > IP midchain out of Tunnel512 7FBE5C6B8B18
              label 123116
              TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
        < 8 > IP midchain out of Tunnel512 7FBE5C6B8B18
              label 123116
              TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
        < 9 > IP midchain out of Tunnel512 7FBE5C6B8B18
              label 123116
              TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
        <10 > IP midchain out of Tunnel512 7FBE5C6B8B18
              label 123116
              TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
        <11 > IP midchain out of Tunnel512 7FBE5C6B8B18
              label 123116
              TAG adj out of GigabitEthernet3, addr 12.39.15.1 7FBE5C6B8EF8
      Subblocks:
        None
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 10.2: Configuring MPLS TE load balancing with IOS XR

We'll repeat the exercise, this time starting from SP2R4 and building two tunnels toward SP2R5 – one via R1 (blue) and one via R3 (red):




```
RP/0/0/CPU0:SP2R4#conf t
RP/0/0/CPU0:SP2R4(config)#interface tunnel-te 415
RP/0/0/CPU0:SP2R4(config-if)#ipv4 unnumbered loopback 0
RP/0/0/CPU0:SP2R4(config-if)#destination 12.39.5.5
RP/0/0/CPU0:SP2R4(config-if)#path-option 10 explicit name TE415
RP/0/0/CPU0:SP2R4(config-if)#autoroute announce
RP/0/0/CPU0:SP2R4(config)#interface tunnel-te 435
RP/0/0/CPU0:SP2R4(config-if)#ipv4 unnumbered loopback 0
RP/0/0/CPU0:SP2R4(config-if)#destination 12.39.5.5
RP/0/0/CPU0:SP2R4(config-if)#path-option 10 explicit name TE435
RP/0/0/CPU0:SP2R4(config-if)#autoroute announce
RP/0/0/CPU0:SP2R4(config-if-tunte-aa)#exit
RP/0/0/CPU0:SP2R4(config-if)#exit
RP/0/0/CPU0:SP2R4(config-expl-path)#index 10 next-address 12.39.1.1
RP/0/0/CPU0:SP2R4(config-expl-path)#index 20 next-address 12.39.5.5
RP/0/0/CPU0:SP2R4(config-expl-path)#explicit-path name TE435
RP/0/0/CPU0:SP2R4(config-expl-path)#index 10 next-address 12.39.3.3
RP/0/0/CPU0:SP2R4(config-expl-path)#index 20 next-address 12.39.5.5
RP/0/0/CPU0:SP2R4(config-expl-path)#commit
```

If everything was configured correctly, the path towards 12.39.5.5 should be balanced 1:1:

```
RP/0/0/CPU0:SP2R4#sh route ipv4 unicast 12.39.5.5

Routing entry for 12.39.5.5/32
  Known via "isis AS1239", distance 115, metric 30, type level-2
  Installed Jan 01 11:01:01.429 for 00:00:18
  Routing Descriptor Blocks
    12.39.5.5, from 12.39.5.5, via tunnel-te415
      Route metric is 30
    12.39.5.5, from 12.39.5.5, via tunnel-te435
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Route metric is 30
  No advertising protos.
```

We can also verify this with CEF:

```
RP/0/0/CPU0:SP2R4#sh cef 12.39.5.5 detail | b Load
Mon Jan 01 11:02:13.454 UTC
    Load distribution: 0 1 (refcount 3)

    Hash  OK  Interface                 Address
    0     Y   tunnel-te415              point2point
    1     Y   tunnel-te435              point2point
```

Unfortunately, IOS XRv currently doesn't support unequal load sharing on TE tunnels. This feature is supported on IOS XRv9000 and hardware platforms, but this is outside of the current exam scope.

The configuration would be however identical to that from IOS XE – using keyword '**load-share**' under interface te415 and te435 definition to adjust the weighting.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 13: MPLS Quality of Service

As currently no QoS features are supported for data-plane in XRv, we will focus configuration on IOS-XE nodes and only configurate control-plane part on XRv for illustration.

While Task 1 is continuation of the previous Chapter and uses existing configuration, from Task 2 until end of this Chapter we'll use slightly modified topology and configuration.

This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch13.virl1.virl`
- **For VIRL²/CML²:** `sp.bootcamp.ch13.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch13.unl`

Task 1: MPLS QoS DS-TE

Task 1.1: DS-TE configuration – RDM model

We'll focus the configuration on IETF (not pre-standard) RDM configuration model. Configuring this model starts with informing MPLS TE feature that we're willing to use IETF RDM model:

On SP1R1-SP1R4 (IOS XR):

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#mpls traffic-eng
RP/0/0/CPU0:SP1R4(config-mpls-te)#ds-te mode ietf
RP/0/0/CPU0:SP1R4(config-mpls-te)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
RP/0/0/CPU0:Jan 1 20:23:49.825 : te_control[1050]: %ROUTING-MPLS_TE-4-WARN : Please
ensure that DS-TE mode is identical on all nodes and trigger link flooding on all nodes
to update TE topology
```

On SP1R5 (IOS-XE):

```
SP1-R5-RR(config)#mpls traffic-eng ds-te mode ietf
```

RDM model allows for sharing bandwidth between up to 8 defined CTs (class-types). IETF RDM model assumes, each CT can further contain up to 8 priorities.

The original (pre-standard) implementation and the CLI assumes we only have two pools: global, usually used for all traffic that's not specifically classified, and sub-pool, that will usually carry high-priority traffic.

Let's adjust SP1R1-SP1R4 configuration to account of 1Gbps of sub-pool out of the defined 10Gbps bandwidth defined already per interface:

```
RP/0/0/CPU0:SP1R1#sh running-config rsvp
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
rsvp
 interface GigabitEthernet0/0/0/0
  bandwidth 10000000
 !
 interface GigabitEthernet0/0/0/2
  bandwidth 10000000
 !
 interface GigabitEthernet0/0/0/3
  bandwidth 10000000
 !
 interface GigabitEthernet0/0/0/5
  bandwidth 10000000
 !
!
```

We're going to add sub-pool in IETF RDM model:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#rsvp
RP/0/0/CPU0:SP1R1(config-rsvp)#interface gigabitEthernet 0/0/0/0
RP/0/0/CPU0:SP1R1(config-rsvp-if)#bandwidth rdm bc0 10 Gbps bc1 1 Gbps
RP/0/0/CPU0:SP1R1(config-rsvp-if)#interface gigabitEthernet 0/0/0/2
RP/0/0/CPU0:SP1R1(config-rsvp-if)#bandwidth rdm bc0 10 Gbps bc1 1 Gbps
RP/0/0/CPU0:SP1R1(config-rsvp-if)#interface gigabitEthernet 0/0/0/3
RP/0/0/CPU0:SP1R1(config-rsvp-if)#bandwidth rdm bc0 10 Gbps bc1 1 Gbps
RP/0/0/CPU0:SP1R1(config-rsvp-if)#interface gigabitEthernet 0/0/0/5
RP/0/0/CPU0:SP1R1(config-rsvp-if)#bandwidth rdm bc0 10 Gbps bc1 1 Gbps
RP/0/0/CPU0:SP1R1(config-rsvp-if)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

Let's check the new configuration:

```
RP/0/0/CPU0:SP1R1#sh running-config rsvp
rsvp
 interface GigabitEthernet0/0/0/0
  bandwidth rdm bc0 10000000 bc1 1000000
 !
 interface GigabitEthernet0/0/0/2
  bandwidth rdm bc0 10000000 bc1 1000000
 !
 interface GigabitEthernet0/0/0/3
  bandwidth rdm bc0 10000000 bc1 1000000
 !
 interface GigabitEthernet0/0/0/5
  bandwidth rdm bc0 10000000 bc1 1000000
 !
!
```

Repeat this configuration on all other IOS XR nodes within AS3356. Next, add the same configuration to SP1R5:

```
SP1-R5-RR(config-if)#ip rsvp bandwidth rdm bc0 10000000 bc1 1000000
```

Let's check the bandwidth mapping on SP1R4:

```
RP/0/0/CPU0:SP1R4#sh rsvp interface

*: RDM: Default I/F B/W % : 75% [default] (max resv/bc0), 0% [default] (bc1)

INTERFACE: GigabitEthernet0/0/0/0 (ifh=0x0).
 BW (bits/sec): Max=10G. MaxFlow=10G.
                Allocated=2M (0%).
                BC0=10G. BC1=1G.

INTERFACE: GigabitEthernet0/0/0/2 (ifh=0x0).
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
 BW (bits/sec): Max=10G. MaxFlow=10G.
                Allocated=2M (0%).
                BC0=10G. BC1=1G.

INTERFACE: GigabitEthernet0/0/0/5 (ifh=0x0).
 BW (bits/sec): Max=10G. MaxFlow=10G.
                Allocated=0 (0%).
                BC0=10G. BC1=1G.
```

We have 2Mbps of bandwidth allocated over Gi0/0/0/0 and Gi0/0/0/2 because we already have some tunnels running over those interfaces:

```
RP/0/0/CPU0:SP1R4#sh mpls traffic-eng tunnels | utility egrep "Name:|Bandwidth"
Name: tunnel-te45  Destination: 33.56.5.5  Ifhandle:0xf0
  Signalled-Name: SP1R4_t45
    Bandwidth Requested: 1000 kbps  CT0
    Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
Name: tunnel-te46  Destination: 33.56.5.5  Ifhandle:0x110
  Signalled-Name: SP1R4_t46
    Bandwidth Requested: 1000 kbps  CT0
    Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
Name: tunnel-te47  Destination: 33.56.5.5  Ifhandle:0x130
  Signalled-Name: SP1R4_t47
    Bandwidth Requested: 1000 kbps  CT0
    Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
  Tunnel Name: SP1R2_t2432 Tunnel Role: Mid
    Bandwidth: 1000 kbps (CT0) Priority:  7  7 DSTE-class: 0
```

Currently, all tunnels request bandwidth from global pool - CT0, which is default configuration. Let's change TE tunnel 45 and 46 to request bandwidth from the sub-pool:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#interface tunnel-te45
RP/0/0/CPU0:SP1R4(config-if)#no signalled-bandwidth 1000
RP/0/0/CPU0:SP1R4(config-if)#signalled-bandwidth sub-pool 1000
RP/0/0/CPU0:SP1R4(config-if)#interface tunnel-te46
RP/0/0/CPU0:SP1R4(config-if)#no signalled-bandwidth 1000
RP/0/0/CPU0:SP1R4(config-if)#signalled-bandwidth sub-pool 1000
RP/0/0/CPU0:SP1R4(config-if)#commit
RP/0/0/CPU0:SP1R4(config-if)#
```

We can verify the change in allocation:

```
RP/0/0/CPU0:SP1R4#sh mpls traffic-eng tunnels | utility egrep "Name:|Bandwidth"
Name: tunnel-te45  Destination: 33.56.5.5  Ifhandle:0xf0
  Signalled-Name: SP1R4_t45
    Bandwidth Requested: 1000 kbps  CT1
    Bandwidth:     1000 kbps (CT1) Priority:  7  7 Affinity: 0x0/0xffff
Name: tunnel-te46  Destination: 33.56.5.5  Ifhandle:0x110
  Signalled-Name: SP1R4_t46
    Bandwidth Requested: 1000 kbps  CT1
    Bandwidth:     1000 kbps (CT1) Priority:  7  7 Affinity: 0x0/0xffff
Name: tunnel-te47  Destination: 33.56.5.5  Ifhandle:0x130
  Signalled-Name: SP1R4_t47
    Bandwidth Requested: 1000 kbps  CT0
    Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
  Tunnel Name: SP1R2_t2432 Tunnel Role: Mid
    Bandwidth: 1000 kbps (CT0) Priority:  7  7 DSTE-class: 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

This however still maps only global/sub-pool traffic per tunnel. Let's check all other mappings:

```
RP/0/0/CPU0:SP1R4#sh mpls traffic-eng ds-te te-class
te-class 0: class-type 0 priority 7 status default
te-class 1: class-type 1 priority 7 status default
te-class 2: unused
te-class 3: unused
te-class 4: class-type 0 priority 0 status default
te-class 5: class-type 1 priority 0 status default
te-class 6: unused
te-class 7: unused
```

We can also verify the TE tunnels 45, 46 and 47 after the reconfiguration:

```
RP/0/0/CPU0:SP1R4#sh mpls traffic-eng topology 33.56.5.5

IGP Id: 33.56.5.5, MPLS TE Id: 33.56.5.5 Router Node  (OSPF AS3356 area 0)
  Link[0]:Point-to-Point, Nbr IGP Id:33.56.3.3, Nbr Node Id:3, gen:9314
      Frag Id:2, Intf Address:33.56.35.5, Intf Id:0
      Nbr Intf Address:33.56.35.3, Nbr Intf Id:0
      TE Metric:1, IGP Metric:1
      Attribute Flags: 0x0
      Ext Admin Group:
          Length: 256 bits
          Value : 0x::
      Attribute Names:
      Switching Capability:None, Encoding:unassigned
      BC Model ID:RDM
      Physical BW:1000000 (kbps), Max Reservable BW:10000000 (kbps)
      BC0:10000000 (kbps) BC1:1000000 (kbps)
                     Total Allocated   Reservable
                     BW (kbps)         BW (kbps)
                     ---------------   -----------
        TE-class[0]:                0        10000000
        TE-class[1]:                0         1000000
        TE-class[2]:                0               0
        TE-class[3]:                0               0
        TE-class[4]:                0        10000000
        TE-class[5]:                0         1000000
        TE-class[6]:                0               0
        TE-class[7]:                0               0

  Link[1]:Point-to-Point, Nbr IGP Id:33.56.1.1, Nbr Node Id:1, gen:9315
      Frag Id:3, Intf Address:33.56.15.5, Intf Id:0
      Nbr Intf Address:33.56.15.1, Nbr Intf Id:0
      TE Metric:1, IGP Metric:1
      Attribute Flags: 0x0
      Ext Admin Group:
          Length: 256 bits
          Value : 0x::
      Attribute Names:
      Switching Capability:None, Encoding:unassigned
      BC Model ID:RDM
      Physical BW:1000000 (kbps), Max Reservable BW:10000000 (kbps)
      BC0:10000000 (kbps) BC1:0 (kbps)
                     Total Allocated   Reservable
                     BW (kbps)         BW (kbps)
                     ---------------   -----------
        TE-class[0]:                0        10000000
        TE-class[1]:                0               0
        TE-class[2]:                0               0
        TE-class[3]:                0               0
        TE-class[4]:                0        10000000
        TE-class[5]:                0               0
        TE-class[6]:                0               0
        TE-class[7]:                0               0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

We can also verify the OSPF LSA opaque LSAs to check if they do carry over the DS-TE information:

```
RP/0/0/CPU0:SP1R4#sh ospf database opaque-area self-originate

            OSPF Router with ID (33.56.4.4) (Process ID AS3356)

                Type-10 Opaque Link Area Link States (Area 0)

[...]

    Link connected to Point-to-Point network
      Link ID : 33.56.1.1
[...]
      Maximum bandwidth : 125000000
      Maximum reservable bandwidth global: 1250000000
      DS-TE Available bandwidth (bytes/sec):
      TE Class 0 :           1249749888  TE Class 1 :            124875000
      TE Class 2 :                    0  TE Class 3 :                    0
      TE Class 4 :           1250000000  TE Class 5 :            125000000
      TE Class 6 :                    0  TE Class 7 :                    0
      Affinity Bit : 0
      IGP Metric : 1
      DS-TE Bandwidth Constraints :
        Allocation Model : Not set (0)
        Number of Bandwidth Constraints : 2
        Constraint 0 :           1250000000
        Constraint 1 :            125000000

Link connected to Point-to-Point network
      Link ID : 33.56.3.3
[...]
      Maximum bandwidth : 125000000
      Maximum reservable bandwidth global: 1250000000
      DS-TE Available bandwidth (bytes/sec):
      TE Class 0 :           1249749888  TE Class 1 :            124875000
      TE Class 2 :                    0  TE Class 3 :                    0
      TE Class 4 :           1250000000  TE Class 5 :            125000000
      TE Class 6 :                    0  TE Class 7 :                    0
      Affinity Bit : 0
      IGP Metric : 1
      DS-TE Bandwidth Constraints :
        Allocation Model : Not set (0)
        Number of Bandwidth Constraints : 2
        Constraint 0 :           1250000000
        Constraint 1 :            125000000
[...]
```

Let's configure non-default CT mapping for TE class:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#mpls traffic-eng
RP/0/0/CPU0:SP1R1(config-mpls-te)#ds-te te-classes
RP/0/0/CPU0:SP1R1(config-te-class)#te-class 7 class-type 1 priority 5
```

Let's review current TE-CT mapping:

```
RP/0/0/CPU0:SP1R1#sh mpls traffic-eng ds-te te-class
te-class 0: class-type 0 priority 7 status default
te-class 1: class-type 1 priority 7 status default
te-class 2: unused
te-class 3: unused
te-class 4: class-type 0 priority 0 status default
te-class 5: class-type 1 priority 0 status default
te-class 6: unused
te-class 7: class-type 1 priority 5 status configured
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

And check the interface MPLS TE mapping on the links:

```
RP/0/0/CPU0:SP1R1#sh mpls traffic-eng topology 33.56.5.5

IGP Id: 33.56.5.5, MPLS TE Id: 33.56.5.5 Router Node  (OSPF AS3356 area 0)

  Link[0]:Point-to-Point, Nbr IGP Id:33.56.3.3, Nbr Node Id:3, gen:9566
      Frag Id:2, Intf Address:33.56.35.5, Intf Id:0
      Nbr Intf Address:33.56.35.3, Nbr Intf Id:0
      TE Metric:1, IGP Metric:1
      Attribute Flags: 0x0
      Ext Admin Group:
          Length: 256 bits
          Value : 0x::
      Attribute Names:
      Switching Capability:None, Encoding:unassigned
      BC Model ID:RDM
      Physical BW:1000000 (kbps), Max Reservable BW:10000000 (kbps)
      BC0:10000000 (kbps) BC1:1000000 (kbps)
                      Total Allocated   Reservable
                      BW (kbps)         BW (kbps)
                      ---------------   -----------
        TE-class[0]:             0       10000000
        TE-class[1]:             0        1000000
        TE-class[2]:             0              0
        TE-class[3]:             0              0
        TE-class[4]:             0       10000000
        TE-class[5]:             0        1000000
        TE-class[6]:             0              0
        TE-class[7]:       1000000              0
```

Please note, that to properly use the new mapping, it has to be extended to all nodes within administrative domain (AS 3356 in this case).

Selection of traffic within TE tunnels is made with adding ‘class-type’ to the tunnel bandwidth specification request, like that:

```
RP/0/0/CPU0:SP1R1(config)#interface tunnel-te 510
RP/0/0/CPU0:SP1R1(config-if)#signalled-bandwidth 1000 class-type 0
 ! example, low-priority tunnel
RP/0/0/CPU0:SP1R1(config)#interface tunnel-te 520
RP/0/0/CPU0:SP1R1(config-if)#signalled-bandwidth 330 class-type 1
 ! example, high-priority tunnel
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: DS-TE configuration – MAM model

We will migrate the configuration to MAM model. MAM model doesn't allow for sharing the bandwidth between traffic classes.

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#mpls traffic-eng
RP/0/0/CPU0:SP1R1(config-mpls-te)#ds-te bc-model mam
RP/0/0/CPU0:SP1R1(config-mpls-te)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

Let's modify the interface configuration within RSVP to account for the MAM model:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#rsvp
RP/0/0/CPU0:SP1R1(config-rsvp)#interface GigabitEthernet0/0/0/0
RP/0/0/CPU0:SP1R1(config-rsvp-if)#bandwidth mam max-reservable-bw 10 Gbps bc0 8 Gbps bc1
2 Gbps
RP/0/0/CPU0:SP1R1(config-rsvp-if)#no bandwidth rdm
RP/0/0/CPU0:SP1R1(config-rsvp-if)#interface gi0/0/0/2
RP/0/0/CPU0:SP1R1(config-rsvp-if)#bandwidth mam max-reservable-bw 10 Gbps bc0 8 Gbps bc1
2 Gbps
RP/0/0/CPU0:SP1R1(config-rsvp-if)#no bandwidth rdm
RP/0/0/CPU0:SP1R1(config-rsvp-if)#interface gi0/0/0/3
RP/0/0/CPU0:SP1R1(config-rsvp-if)#bandwidth mam max-reservable-bw 10 Gbps bc0 8 Gbps bc1
2 Gbps
RP/0/0/CPU0:SP1R1(config-rsvp-if)#no bandwidth rdm
RP/0/0/CPU0:SP1R1(config-rsvp-if)#interface gi0/0/0/5
RP/0/0/CPU0:SP1R1(config-rsvp-if)#bandwidth mam max-reservable-bw 10 Gbps bc0 8 Gbps bc1
2 Gbps
RP/0/0/CPU0:SP1R1(config-rsvp-if)#no bandwidth rdm
RP/0/0/CPU0:SP1R1(config-rsvp-if)#
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

The clean configuration should look like this:

```
RP/0/0/CPU0:SP1R1#sh running-config rsvp
rsvp
 interface GigabitEthernet0/0/0/0
  bandwidth mam max-reservable-bw 10000000 bc0 8000000 bc1 2000000
 !
 interface GigabitEthernet0/0/0/2
  bandwidth mam max-reservable-bw 10000000 bc0 8000000 bc1 2000000
 !
 interface GigabitEthernet0/0/0/3
  bandwidth mam max-reservable-bw 10000000 bc0 8000000 bc1 2000000
 !
 interface GigabitEthernet0/0/0/5
  bandwidth mam max-reservable-bw 10000000 bc0 8000000 bc1 2000000
 !
!
```

Now, let's check the current state of RSVP and MPLS TE information:

```
RP/0/0/CPU0:SP1R1#sh mpls traffic-eng topology 33.56.1.1

IGP Id: 33.56.1.1, MPLS TE Id: 33.56.1.1 Router Node  (OSPF AS3356 area 0)

  Link[0]:Point-to-Point, Nbr IGP Id:33.56.4.4, Nbr Node Id:4, gen:9836
      Frag Id:4, Intf Address:33.56.14.1, Intf Id:0
      Nbr Intf Address:33.56.14.4, Nbr Intf Id:0
      TE Metric:1, IGP Metric:1
      Attribute Flags: 0x0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    Ext Admin Group:
        Length: 256 bits
        Value : 0x::
    Attribute Names:
    Switching Capability:None, Encoding:unassigned
    BC Model ID:MAM
    Physical BW:1000000 (kbps), Max Reservable BW:10000000 (kbps)
    BC0:8000000 (kbps) BC1:2000000 (kbps)
                    Total Allocated   Reservable
                    BW (kbps)         BW (kbps)
                    ---------------   -----------
      TE-class[0]:                0         8000000
      TE-class[1]:                0         2000000
      TE-class[2]:                0               0
      TE-class[3]:                0               0
      TE-class[4]:                0         8000000
      TE-class[5]:                0         2000000
      TE-class[6]:                0               0
      TE-class[7]:                0         2000000

Link[1]:Point-to-Point, Nbr IGP Id:33.56.2.2, Nbr Node Id:2, gen:9837
    Frag Id:6, Intf Address:33.56.12.1, Intf Id:0
    Nbr Intf Address:33.56.12.2, Nbr Intf Id:0
    TE Metric:1, IGP Metric:1
    Attribute Flags: 0x0
    Ext Admin Group:
        Length: 256 bits
        Value : 0x::
    Attribute Names:
    Switching Capability:None, Encoding:unassigned
    BC Model ID:MAM
    Physical BW:1000000 (kbps), Max Reservable BW:10000000 (kbps)
    BC0:8000000 (kbps) BC1:2000000 (kbps)
                    Total Allocated   Reservable
                    BW (kbps)         BW (kbps)
                    ---------------   -----------
      TE-class[0]:                0         8000000
      TE-class[1]:                0         2000000
      TE-class[2]:                0               0
      TE-class[3]:                0               0
      TE-class[4]:                0         8000000
      TE-class[5]:                0         2000000
      TE-class[6]:                0               0
      TE-class[7]:                0         2000000

Link[2]:Point-to-Point, Nbr IGP Id:33.56.5.5, Nbr Node Id:5, gen:9838
    Frag Id:7, Intf Address:33.56.15.1, Intf Id:0
    Nbr Intf Address:33.56.15.5, Nbr Intf Id:0
    TE Metric:1, IGP Metric:1
    Attribute Flags: 0x0
    Ext Admin Group:
        Length: 256 bits
        Value : 0x::
    Attribute Names:
    Switching Capability:None, Encoding:unassigned
    BC Model ID:MAM
    Physical BW:1000000 (kbps), Max Reservable BW:10000000 (kbps)
    BC0:8000000 (kbps) BC1:2000000 (kbps)
                    Total Allocated   Reservable
                    BW (kbps)         BW (kbps)
                    ---------------   -----------
      TE-class[0]:             1001         7998999
      TE-class[1]:                0         2000000
      TE-class[2]:                0               0
      TE-class[3]:                0               0
      TE-class[4]:                0         8000000
      TE-class[5]:                0         2000000
      TE-class[6]:                0               0
      TE-class[7]:                0         2000000
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Link[3]:Point-to-Point, Nbr IGP Id:33.56.3.3, Nbr Node Id:3, gen:9839
    Frag Id:9, Intf Address:33.56.13.1, Intf Id:0
    Nbr Intf Address:33.56.13.3, Nbr Intf Id:0
    TE Metric:1, IGP Metric:1
    Attribute Flags: 0x0
    Ext Admin Group:
        Length: 256 bits
        Value : 0x::
    Attribute Names:
    Switching Capability:None, Encoding:unassigned
    BC Model ID:MAM
    Physical BW:1000000 (kbps), Max Reservable BW:10000000 (kbps)
    BC0:8000000 (kbps) BC1:2000000 (kbps)
                   Total Allocated   Reservable
                   BW (kbps)         BW (kbps)
                   ---------------   -----------
      TE-class[0]:            0        8000000
      TE-class[1]:            0        2000000
      TE-class[2]:            0              0
      TE-class[3]:            0              0
      TE-class[4]:            0        8000000
      TE-class[5]:            0        2000000
      TE-class[6]:            0              0
      TE-class[7]:            0        2000000
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3: CBTS – IOS only

Class Based Tunnel Selection is an IOS/IOS-XE feature, which selects outgoing TE tunnel based on the EXP/TC field value.

CBTS requires multiple tunnels to the same destination, and configuration of master tunnel than will then sub-route traffic to specific child tunnels based on the EXP/TC marking. In below configuration:

- all traffic with EXP/TC value of 0 will go via Tunnel10
- all traffic with EXP/TC value of 1 will go via Tunnel20
- all traffic with all other EXP/TC values will go via Tunnel30

```
SP1-R5-RR#conf t
SP1-R5-RR(config)#interface tunnel10
SP1-R5-RR(config-if)#ip unnumbered lo0
SP1-R5-RR(config-if)#tunnel mode mpls traffic-eng
SP1-R5-RR(config-if)#tunnel destination 33.56.2.2
SP1-R5-RR(config-if)#tunnel mpls traffic-eng autoroute announce
SP1-R5-RR(config-if)#tunnel mpls traffic-eng path-option 10 dynamic
SP1-R5-RR(config-if)#tunnel mpls traffic-eng exp 0
SP1-R5-RR(config-if)#^Z
SP1-R5-RR(config)#int tun20
SP1-R5-RR(config-if)#ip unnumbered Loopback0
SP1-R5-RR(config-if)#tunnel mode mpls traffic-eng
SP1-R5-RR(config-if)#tunnel destination 33.56.2.2
SP1-R5-RR(config-if)#tunnel mpls traffic-eng autoroute announce
SP1-R5-RR(config-if)#tunnel mpls traffic-eng path-option 10 dynamic
SP1-R5-RR(config-if)#tunnel mpls traffic-eng exp 1
SP1-R5-RR(config-if)#int tu30
SP1-R5-RR(config-if)#ip unnumbered Loopback0
SP1-R5-RR(config-if)#tunnel mode mpls traffic-eng
SP1-R5-RR(config-if)#tunnel destination 33.56.2.2
SP1-R5-RR(config-if)#tunnel mpls traffic-eng path-option 10 dynamic
SP1-R5-RR(config-if)#tunnel mpls traffic-eng autoroute announce
SP1-R5-RR(config-if)#tunnel mpls traffic-eng exp default
SP1-R5-RR(config)#interface tunnel444
SP1-R5-RR(config-if)#ip unnumbered Loopback0
SP1-R5-RR(config-if)#tunnel mode mpls traffic-eng
SP1-R5-RR(config-if)#tunnel destination 33.56.2.2
SP1-R5-RR(config-if)#tunnel mpls traffic-eng autoroute announce
SP1-R5-RR(config-if)#tunnel mpls traffic-eng path-option 10 dynamic
SP1-R5-RR(config-if)#tunnel mpls traffic-eng exp-bundle master
SP1-R5-RR(config-if)#tunnel mpls traffic-eng exp-bundle member tunnel 10
SP1-R5-RR(config-if)#tunnel mpls traffic-eng exp-bundle member tunnel 20
SP1-R5-RR(config-if)#tunnel mpls traffic-eng exp-bundle member tunnel 30
SP1-R5-RR(config-if)#^Z
```

The route table should point all three tunnels to the same destination as per configuration:

```
SP1-R5-RR#sh ip route 33.56.2.2
Routing entry for 33.56.2.2/32
  Known via "ospf 100", distance 110, metric 3, type intra area
  Last update from 33.56.2.2 on Tunnel10, 00:00:04 ago
  Routing Descriptor Blocks:
    33.56.2.2, from 33.56.2.2, 00:00:04 ago, via Tunnel10
      Route metric is 3, traffic share count is 1
    33.56.2.2, from 33.56.2.2, 00:00:04 ago, via Tunnel20
      Route metric is 3, traffic share count is 1
  * 33.56.2.2, from 33.56.2.2, 00:00:04 ago, via Tunnel30
      Route metric is 3, traffic share count is 1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

We can also check the CBTS configuration for the node:

```
SP1-R5-RR#sh mpls traffic-eng exp

Destination: 33.56.2.2

   Master: Tunnel444            Status: up

   Members         Status               Conf Exp        Actual Exp
   Tunnel10        up  (Active)         0               0
   Tunnel20        up  (Active)         1               1
   Tunnel30        up  (Active)         Default         2 3 4 5 6 7


(D) : Destination is different
(NE): Exp values not configured on tunnel
SP1-R5-RR#
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4: PBTS - IOS XR only

Configure PBTS this way:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#interface tunnel-te 45
RP/0/0/CPU0:SP1R4(config-if)#ipv4 unnumbered loopback 0
RP/0/0/CPU0:SP1R4(config-if)#description R4->R5 tunnel
RP/0/0/CPU0:SP1R4(config-if)#autoroute announce
RP/0/0/CPU0:SP1R4(config-if-tunte-aa)#exit
RP/0/0/CPU0:SP1R4(config-if)#destination 33.56.5.5
RP/0/0/CPU0:SP1R4(config-if)#record-route
RP/0/0/CPU0:SP1R4(config-if)#signalled-bandwidth 1000
RP/0/0/CPU0:SP1R4(config-if)#path-option 10 dynamic
RP/0/0/CPU0:SP1R4(config-if)#policy-class 1 ! this tunnel will serve traffic with
EXP/TC=1
RP/0/0/CPU0:SP1R4(config-if)#commit

RP/0/0/CPU0:SP1R4(config)#interface tunnel-te46
RP/0/0/CPU0:SP1R4(config-if)#description R4->R5 tunnel
RP/0/0/CPU0:SP1R4(config-if)#ipv4 unnumbered Loopback0
RP/0/0/CPU0:SP1R4(config-if)#signalled-bandwidth 1000
RP/0/0/CPU0:SP1R4(config-if)#autoroute announce
RP/0/0/CPU0:SP1R4(config-if-tunte-aa)# destination 33.56.5.5
RP/0/0/CPU0:SP1R4(config-if)#policy-class 2 ! this tunnel will serve traffic with
EXP/TC=2
RP/0/0/CPU0:SP1R4(config-if)#record-route
RP/0/0/CPU0:SP1R4(config-if)#path-option 10 dynamic
RP/0/0/CPU0:SP1R4(config-if)#commit

RP/0/0/CPU0:SP1R4(config)#interface tunnel-te47
RP/0/0/CPU0:SP1R4(config-if)#description R4->R5 tunnel
RP/0/0/CPU0:SP1R4(config-if)#ipv4 unnumbered Loopback0
RP/0/0/CPU0:SP1R4(config-if)#signalled-bandwidth 1000
RP/0/0/CPU0:SP1R4(config-if)#autoroute announce
RP/0/0/CPU0:SP1R4(config-if-tunte-aa)#destination 33.56.5.5
RP/0/0/CPU0:SP1R4(config-if)#policy-class default ! this tunnel will serve all other
traffic
RP/0/0/CPU0:SP1R4(config-if)#record-route
RP/0/0/CPU0:SP1R4(config-if)#path-option 10 dynamic
RP/0/0/CPU0:SP1R4(config-if)#commit
```



Please note that XRv doesn't support policy based routing, so using PBTS with forward-classes is not possible at this time. While in above configuration we're matching EXP bits on three tunnels, the PBR driven PBTS configuration would look the same, with '**forward-class X**' instead of '**policy-class X**' in the tunnel definition, and with addition of PBR configuration on router input interfaces, matching traffic criteria (for example – DSCP or TC information) and setting correct forwarding-class, like this:

```
class-map type traffic match-any cm-voice
 match mpls experimental topmost 5
 match ip dscp ef

class-map type traffic match-any cm-data
 match mpls experimental topmost 0 1 2 3 4

policy-map type pbr pm-PBTS
 class type traffic cm-voice
  set forward-class 5
 class type traffic cm-data
  set forward class 1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński



```
interface te0/1/0/0
 service-policy type pbr input pm-PBTS
```

Current XRv and XRv 9000 implementations (6.0.1/6.1.2) don't support '**type pbr**' for the service policies attached to the interface.

There are three tunnels:

```
RP/0/0/CPU0:SP1R4#sh mpls forwarding prefix 33.56.5.5/32
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
104003 Pop         33.56.5.5/32      tt45         33.56.5.5       5417
       Pop         33.56.5.5/32      tt46         33.56.5.5       0
       Pop         33.56.5.5/32      tt100        33.56.5.5       0
RP/0/0/CPU0:SP1R4#
```

You can verify:

```
RP/0/0/CPU0:SP1R4#sh mpls traffic-eng tunnels property pbts tabular

            Tunnel   LSP      Destination          Source     Tun     FRR  LSP  Path
              Name    ID         Address         Address  State  State Role  Prot
----------------- ----- --------------- --------------- ------ ------ ---- -----
      tunnel-te45     2       33.56.5.5       33.56.4.4     up  Inact Head Inact
      tunnel-te46     2       33.56.5.5       33.56.4.4     up  Inact Head Inact
     tunnel-te100     2       33.56.5.5       33.56.4.4     up  Inact Head Inact

RP/0/0/CPU0:SP1R4#sh mpls traffic-eng tunnels property pbts detail
Name: tunnel-te45  Destination: 33.56.5.5  Ifhandle:0x1b0
  Signalled-Name: SP1R4_t45
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type dynamic  (Basis for Setup, path weight 2)
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 1000 kbps  CT0
    Creation Time: Tue Jan  1 21:46:21 2017 (00:44:58 ago)
  Config Parameters:
    Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: 1
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Not Enabled
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  SNMP Index: 18
  Binding SID: None
  History:
    Tunnel has been up for: 00:44:58 (since Tue Jan 01 21:46:21 UTC 2017)
    Current LSP:
      Uptime: 00:44:58 (since Tue Jan 01 21:46:21 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, identical to the [CURRENT] LSP
        Date/Time: Tue Jan 01 22:28:32 UTC 2017 [00:02:47 ago]
  Current LSP Info:
    Instance: 2, Signaling Area: OSPF AS3356 area 0
    Uptime: 00:44:58 (since Tue Jan 01 21:46:21 UTC 2017)
    Outgoing Interface: GigabitEthernet0/0/0/2, Outgoing Label: 103025
    Router-IDs: local      33.56.4.4
                downstream 33.56.3.3
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Outgoing:
        Explicit Route:
          Strict, 33.56.34.3
          Strict, 33.56.35.5
          Strict, 33.56.5.5

      Record Route: Empty
      Tspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
      Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
    Resv Info:
      Record Route:
        IPv4 33.56.34.3, flags 0x0
        IPv4 33.56.35.5, flags 0x0
      Fspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
  Persistent Forwarding Statistics:
    Out Bytes: 115065
    Out Packets: 773

Name: tunnel-te46  Destination: 33.56.5.5  Ifhandle:0x1d0
  Signalled-Name: SP1R4_t46
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type dynamic  (Basis for Setup, path weight 2)
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 1000 kbps  CT0
    Creation Time: Tue Jan  1 21:49:25 2017 (00:41:54 ago)
  Config Parameters:
    Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: 2
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  SNMP Index: 23
  Binding SID: None
  History:
    Tunnel has been up for: 00:41:54 (since Tue Jan 01 21:49:25 UTC 2017)
    Current LSP:
      Uptime: 00:41:54 (since Tue Jan 01 21:49:25 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, identical to the [CURRENT] LSP
        Date/Time: Tue Jan 01 22:28:32 UTC 2017 [00:02:47 ago]
  Current LSP Info:
    Instance: 2, Signaling Area: OSPF AS3356 area 0
    Uptime: 00:41:54 (since Tue Jan 01 21:49:25 UTC 2017)
    Outgoing Interface: GigabitEthernet0/0/0/0, Outgoing Label: 101009
    Router-IDs: local      33.56.4.4
                downstream 33.56.1.1
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Outgoing:
        Explicit Route:
          Strict, 33.56.14.1
          Strict, 33.56.15.5
          Strict, 33.56.5.5

      Record Route: Empty
      Tspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
      Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
    Resv Info:
      Record Route:
        IPv4 33.56.14.1, flags 0x0
        IPv4 33.56.15.5, flags 0x0
      Fspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
  Persistent Forwarding Statistics:
    Out Bytes: 49200
    Out Packets: 500

Name: tunnel-te47  Destination: 33.56.5.5  Ifhandle:0x1f0
  Signalled-Name: SP1R4_t100
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10,  type dynamic  (Basis for Setup, path weight 2)
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 1000 kbps  CT0
    Creation Time: Tue Jan  1 22:23:49 2017 (00:07:30 ago)
  Config Parameters:
    Bandwidth:     1000 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: default
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  SNMP Index: 24
  Binding SID: None
  History:
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    Tunnel has been up for: 00:07:30 (since Tue Jan 01 22:23:49 UTC 2017)
    Current LSP:
      Uptime: 00:07:30 (since Tue Jan 01 22:23:49 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, identical to the [CURRENT] LSP
        Date/Time: Tue Jan 01 22:28:32 UTC 2017 [00:02:47 ago]
  Current LSP Info:
    Instance: 2, Signaling Area: OSPF AS3356 area 0
    Uptime: 00:07:30 (since Tue Jan 01 22:23:49 UTC 2017)
    Outgoing Interface: GigabitEthernet0/0/0/0, Outgoing Label: 101010
    Router-IDs: local      33.56.4.4
                downstream 33.56.1.1
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Outgoing:
        Explicit Route:
          Strict, 33.56.14.1
          Strict, 33.56.15.5
          Strict, 33.56.5.5

      Record Route: Empty
      Tspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
      Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
    Resv Info:
      Record Route:
        IPv4 33.56.14.1, flags 0x0
        IPv4 33.56.15.5, flags 0x0
      Fspec: avg rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
  Persistent Forwarding Statistics:
    Out Bytes: 1176
    Out Packets: 16
Displayed 3 (of 3) heads, 0 (of 0) midpoints, 0 (of 3) tails
Displayed 3 up, 0 down, 0 recovering, 0 recovered heads
```

We can also verify the CEF information, including the EXP/TC mapping:

```
RP/0/0/CPU0:SP1R4#show cef ipv4 33.56.5.5/32
33.56.5.5/32, version 180815, internal 0x1000001 0x0 (ptr 0xa1408a74) [1], 0x0
(0xa13d3f14), 0xa20 (0xa1a2a050)
 Updated Jan 1 20:10:00.321
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 33.56.5.5/32, tunnel-te45, 3 dependencies, weight 0, class 1 [flags 0x0]
    path-idx 0 NHID 0x0 [0xa10e4d78 0xa10e4cd0]
    next hop 33.56.5.5/32
    local adjacency
     local label 104003      labels imposed {ImplNull}
   via 33.56.5.5/32, tunnel-te46, 3 dependencies, weight 0, class 2 [flags 0x0]
    path-idx 1 NHID 0x0 [0xa10e4ec8 0xa10e4e20]
    next hop 33.56.5.5/32
    local adjacency
     local label 104003      labels imposed {ImplNull}
   via 33.56.5.5/32, tunnel-te47, 3 dependencies, weight 0, class default [flags 0x0]
    path-idx 2 NHID 0x0 [0xa10e5018 0xa10e4f70]
    next hop 33.56.5.5/32
    local adjacency
     local label 104003      labels imposed {ImplNull}
```

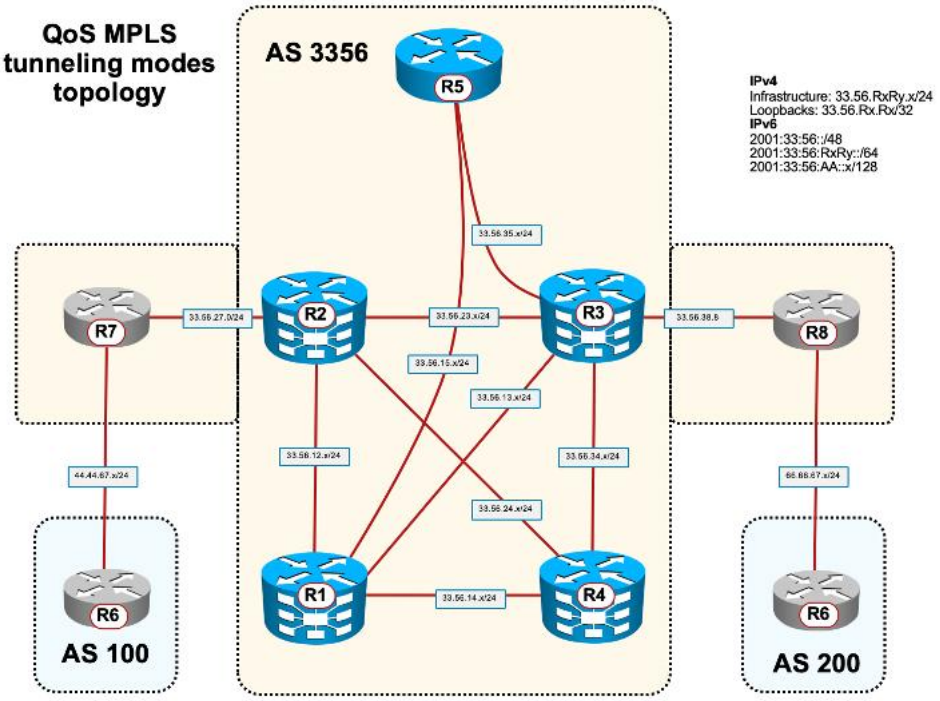

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Service provider MPLS QoS basics

Note, that the topology is simplified a bit for this and next tasks, as we will treat CEL7 (now SP1R7) and CER7 (now SP1R8) as parts of AS3356.

This task until end of chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch13.qos.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch13.qos.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch13-qos.unl`



Subnets were extended between new R7 and R8 routers following the same idea as in the SP cores.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.1: Simple MQC operations on EXP/TC

Cisco MQC on IOS, IOS-XE and IOS XR supports matching the EXP/TC fields within class-maps:

```
CEL7(config)#class-map match-any match-mpls-exp
CEL7(config-cmap)#match mpls experimental topmost 0 1 2 3 4 5 6 7
CEL7(config-cmap)#exit
CEL7(config)#policy-map pm-mpls-exp-match
CEL7(config-pmap)#class  match-mpls-exp
```

Then:

```
CEL7(config)#int gi2
CEL7(config-if)#service-policy output pm-mpls-exp-match
CEL7(config-if)#int gi3.32
CEL7(config-subif)#service-policy output pm-mpls-exp-match
```

And let's initiate some traffic:

```
CEL7#traceroute 66.66.67.6
Type escape sequence to abort.
Tracing the route to 66.66.67.6
VRF info: (vrf in name/id, vrf out name/id)
  1 40.60.27.2 1 msec 2 msec 1 msec
  2 33.56.24.4 [MPLS: Labels 0/104060 Exp 0] 21 msec 20 msec 18 msec
  3 11.11.11.254 [MPLS: Label 123425 Exp 0] 20 msec 22 msec 20 msec
  4 12.39.34.3 [MPLS: Labels 123302/123216 Exp 0] 20 msec 20 msec 20 msec
  5 12.39.23.2 [MPLS: Labels 0/123216 Exp 0] 17 msec 19 msec 19 msec
  6 20.40.27.7 21 msec 18 msec 20 msec
CEL7#sh policy-map interface gigabitEthernet 2
 GigabitEthernet2

  Service-policy output: pm-mpls-exp-match

    Class-map: match-mpls-exp (match-any)
      2 packets, 142 bytes
      5 minute offered rate 0000 bps
      Match: mpls experimental topmost 0  1  2  3  4  5  6  7

      ! note as we're matching all possible EXP/TC values, this entry
      ! matches all possible MPLS packets

    Class-map: class-default (match-any)
      2 packets, 171 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

      ! this entry matches all other traffic - in our case it's IP traffic
      ! unlabelled
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.2: Policing of traffic with specific EXP/TC value

We'll police outgoing traffic with EXP/TC field set to 0 to 8kbps:

```
CEL7(config)#class-map match-exp-0
CEL7(config-cmap)#match mpls experimental topmost 0
CEL7(config-cmap)#exit
CEL7(config)#policy-map pm-police-8kbps
CEL7(config-pmap)#class match-exp-0
CEL7(config-pmap-c)#police 8192
CEL7(config-pmap-c-police)#int gi2
CEL7(config-if)#no service-policy output pm-mpls-exp-match
CEL7(config-if)#service-policy output pm-police-8kbps
CEL7(config-if)#^Z
CEL7#
```

Let's check the result:

```
CEL7#sh policy-map interface gigabitEthernet 2
 GigabitEthernet2
  Service-policy output: pm-police-8kbps
    Class-map: match-exp-0 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: mpls experimental topmost 0
      police:
          cir 8000 bps, bc 1500 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceeded 0000 bps

    Class-map: class-default (match-any)
      16 packets, 1527 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

Then some traffic:

```
CEL7#ping 192.168.7.72
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.7.72, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
CEL7#sh policy-map interface gigabitEthernet 2
 GigabitEthernet2
  Service-policy output: pm-police-8kbps
    Class-map: match-exp-0 (match-all)
      5 packets, 610 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: mpls experimental topmost 0
      police:
          cir 8000 bps, bc 1500 bytes
        conformed 5 packets, 610 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceeded 0000 bps

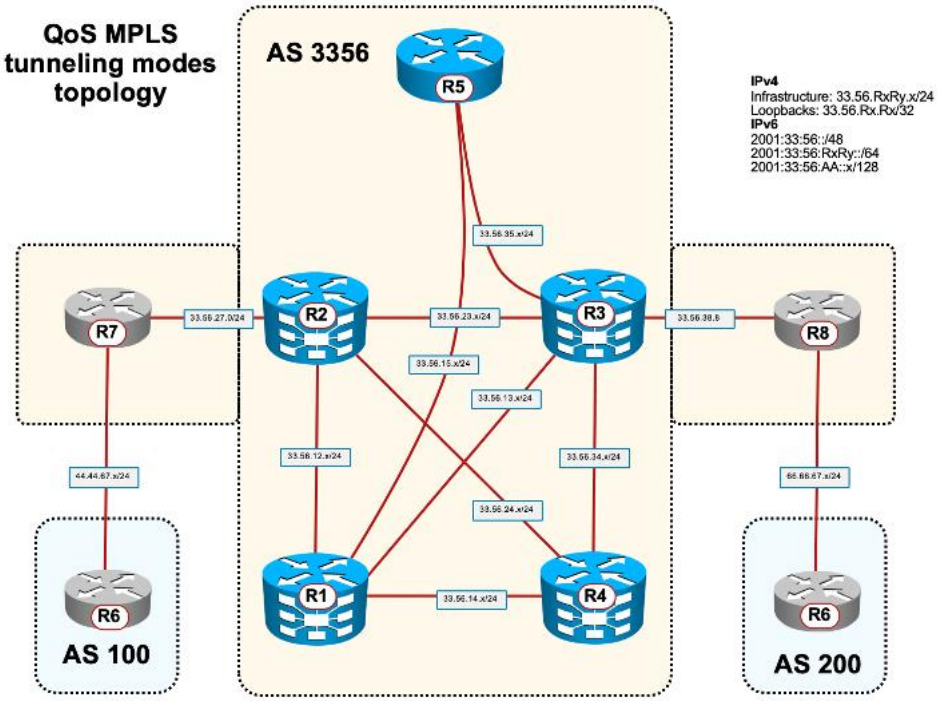
    Class-map: class-default (match-any)
      795 packets, 306575 bytes
      5 minute offered rate 1000 bps, drop rate 0000 bps
      Match: any
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3: Implementing uniform MPLS QoS model

For starters and to get acquainted with MPLS QoS MQC mechanics, we will go with Uniform model.

From RFC 3270, Uniform model defines a unified DiffServ domain, where all nodes conform to the same DiffServ treatment or accept it.



© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.1: Checking probes on AS 100 and AS 200

Both AS 100 R6 and AS 200 R6 are sending IP SLA ICMP probes with specific ToS settings of 1 (DSCP: CS1), 2 (DSCP: CS2) and 3 (DSCP: CS3). They are configured as follows:

CEL6:

```
CEL6#sh running-config | s ip sla
ip sla 101
 icmp-echo 200.200.200.201 source-ip 100.100.100.101
 request-data-size 100
 tos 32
ip sla schedule 101 life forever start-time now
ip sla 102
 icmp-echo 200.200.200.202 source-ip 100.100.100.102
 request-data-size 100
 tos 64
ip sla schedule 102 life forever start-time now
ip sla 103
 icmp-echo 200.200.200.203 source-ip 100.100.100.103
 request-data-size 100
 tos 96
ip sla schedule 103 life forever start-time now
```

CER6:

```
CER6#sh running-config | s ip sla
ip sla 101
 icmp-echo 100.100.100.101 source-ip 200.200.200.201
 request-data-size 100
 tos 32
ip sla schedule 101 life forever start-time now
ip sla 102
 icmp-echo 100.100.100.102 source-ip 200.200.200.202
 request-data-size 100
 tos 64
ip sla schedule 102 life forever start-time now
ip sla 103
 icmp-echo 100.100.100.103 source-ip 200.200.200.203
 request-data-size 100
 tos 96
ip sla schedule 103 life forever start-time now
```

You can and should verify this works before proceeding to Task 3.2, by showing the status:

```
CEL6#sh ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 101
        Latest RTT: 17 milliseconds
Latest operation start time: <timestamp>
Latest operation return code: OK
Number of successes: 23
Number of failures: 0
Operation time to live: Forever

IPSLA operation id: 102
        Latest RTT: 18 milliseconds
Latest operation start time: <timestamp>
Latest operation return code: OK
Number of successes: 23
Number of failures: 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Operation time to live: Forever

IPSLA operation id: 103
        Latest RTT: 17 milliseconds
Latest operation start time: <timestamp>
Latest operation return code: OK
Number of successes: 23
Number of failures: 0
Operation time to live: Forever
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2: Configuring PE SP1R8 router

We'll configure SP1R8 to enable us to monitor EXP markings on the egress of MPLS domain. That way, depending on the marking of probes from CEL6 to CER6 we'll see the change. So, for SP1R8:

```
SP1R8#conf t
SP1R8(config)#class-map mpls-exp0
SP1R8(config-cmap)#match mpls experimental topmost 0
SP1R8(config-cmap)#class-map mpls-exp1
SP1R8(config-cmap)#match mpls experimental topmost 1
SP1R8(config-cmap)#class-map mpls-exp2
SP1R8(config-cmap)#match mpls experimental topmost 2
SP1R8(config-cmap)#class-map mpls-exp3
SP1R8(config-cmap)#match mpls experimental topmost 3
SP1R8(config-cmap)#class-map mpls-exp4
SP1R8(config-cmap)#match mpls experimental topmost 4
SP1R8(config-cmap)#class-map mpls-exp5
SP1R8(config-cmap)#match mpls experimental topmost 5
SP1R8(config-cmap)#class-map mpls-exp6
SP1R8(config-cmap)#match mpls experimental topmost 6
SP1R8(config-cmap)#class-map mpls-exp7
SP1R8(config-cmap)#match mpls experimental topmost 7
SP1R8(config-cmap)#exit
SP1R8(config)#policy-map pm-mpls-core
SP1R8(config-pmap)#class mpls-exp0
SP1R8(config-pmap-c)#class mpls-exp1
SP1R8(config-pmap-c)#class mpls-exp2
SP1R8(config-pmap-c)#class mpls-exp3
SP1R8(config-pmap-c)#class mpls-exp4
SP1R8(config-pmap-c)#class mpls-exp5
SP1R8(config-pmap-c)#class mpls-exp6
SP1R8(config-pmap-c)#class mpls-exp7
SP1R8(config-pmap-c)#exit

SP1R8(config)#int gi2
SP1R8(config-if)#load-interval 30
SP1R8(config-if)#service-policy input pm-mpls-core
```

After a while, policy-map can be checked to verify if the default behaviour of copying the ToS field to MPLS EXP field works. Traffic incoming from AS100:

```
SP1R8#sh policy-map interface gigabitEthernet 2 | i Class|packets,
    Class-map: mpls-exp0 (match-all)
      0 packets, 0 bytes
    Class-map: mpls-exp1 (match-all)
      0 packets, 0 bytes
    Class-map: mpls-exp2 (match-all)
      0 packets, 0 bytes
    Class-map: mpls-exp3 (match-all)
      0 packets, 0 bytes
    Class-map: mpls-exp4 (match-all)
      0 packets, 0 bytes
    Class-map: mpls-exp5 (match-all)
      0 packets, 0 bytes
    Class-map: mpls-exp6 (match-all)
      0 packets, 0 bytes
    Class-map: mpls-exp7 (match-all)
      0 packets, 0 bytes
    Class-map: class-default (match-any)
      22 packets, 2313 bytes
```

What's wrong? class-default sees some packets, but no other classes match? That's because by default, the implicit-null label is being signalled from SP1R3

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

towards SP1R8. Label is popped on last hop – SP1R3, so there's no way we can classify incoming traffic based on EXP/TC field:

```
RP/0/0/CPU0:SP1R3#sh mpls forwarding prefix 33.56.8.8/32
Thu Jan  3 22:00:04.383 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24004  Pop         33.56.8.8/32       Gi0/0/0/1    33.56.38.8      661
```

After reconfiguration on SP1R8:

```
SP1R8#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SP1R8(config)#mpls ldp explicit-null
```

…label is signalled:

```
RP/0/0/CPU0:SP1R3#sh mpls forwarding prefix 33.56.8.8/32
Thu Jan  3 22:01:58.675 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24004  Exp-Null-v4 33.56.8.8/32       Gi0/0/0/1    33.56.38.8      170
```

And we should start to receive EXP/TC marked traffic:

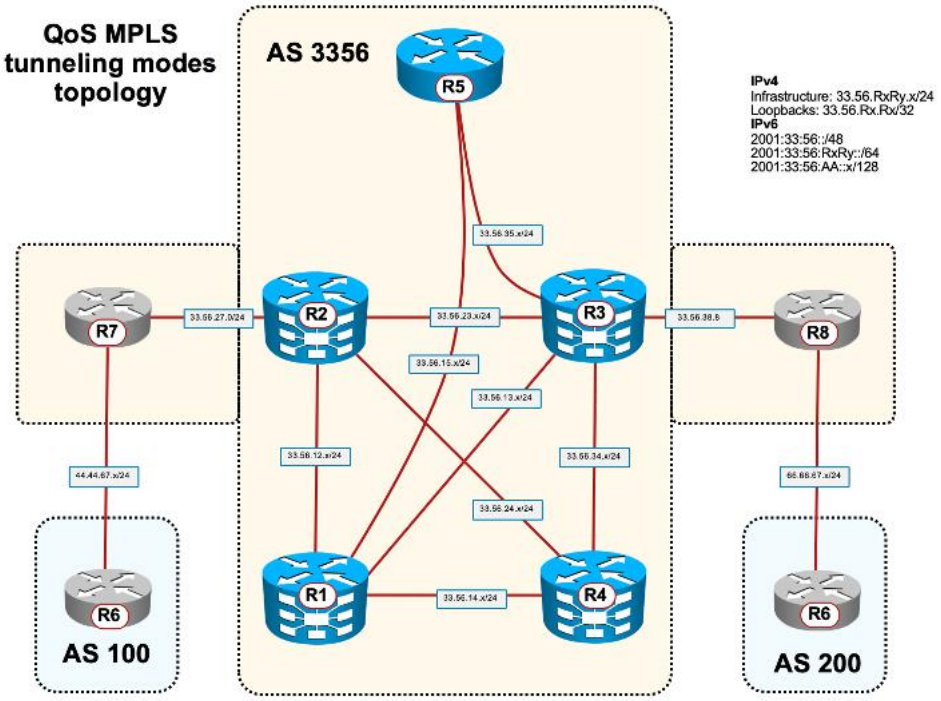
```
SP1R8#sh policy-map interface gigabitEthernet 2 | i Class|packets,
    Class-map: mpls-exp0 (match-all)
      0 packets, 0 bytes
    Class-map: mpls-exp1 (match-all)
      17 packets, 2618 bytes
    Class-map: mpls-exp2 (match-all)
      17 packets, 2618 bytes
    Class-map: mpls-exp3 (match-all)
      17 packets, 2618 bytes
    Class-map: mpls-exp4 (match-all)
      0 packets, 0 bytes
    Class-map: mpls-exp5 (match-all)
      0 packets, 0 bytes
    Class-map: mpls-exp6 (match-all)
      38 packets, 2910 bytes
    Class-map: mpls-exp7 (match-all)
      0 packets, 0 bytes
    Class-map: class-default (match-any)
      356 packets, 33176 bytes
```

As we're dealing with Uniform model, any behaviour on egress is up to us – we can remark the traffic however it fits our needs.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4: Implementing pipe MPLS QoS model

Continuing from the previous task, we'll deploy pipe model for our network.



Task 4.1: Configuring PE routers – SP1R7 and SP1R8

MPLS QoS pipe model specifies, that the queuing behaviour of egress PE should be that queuing should be based on EXP values, not underlying IP ToS/Precedence/DSCP marking.

As we're already signalling explicit-null from SP1R8 towards MPLS cloud, we need to make sure the same happens for SP1R7 and then, configure queuing.

Task 4.1.1: Explicit null signalling from SP1R7

On SP1R7:

```
SP1R7#conf t
SP1R7(config)#mpls ldp explicit-null
```

Check this on SP1R2:

```
RP/0/0/CPU0:SP1R2#sh mpls forwarding prefix 33.56.7.7/32
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
24003  Exp-Null-v4 33.56.7.7/32      Gi0/0/0/1    33.56.27.7      1075
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.1.2: Modifying the EXP values on SP1R7 and SP1R8

For now, we're receiving EXP/TC values of 1, 2 and 3 from the IP SLA probes. This is based on default behaviour of copying the ToS value to MPLS EXP/TC field. To showcase "pipe" mode of typical SP, we're going to modify those default markings towards MPLS core from SP1R7 perspective as well as from SP1R8 perspective to some other mappings.

Let's assume we're going to do as follows for our AS 3356 for any traffic incoming from customers (AS 100 and AS 200):

- the CS1 traffic originally getting EXP value of 1, will be remarked to EXP 4
- the CS2 traffic originally getting EXP value of 2, will be remarked to EXP 4 as well
- the CS3 traffic originally getting EXP value of 3, will be marked down to EXP 0

This requires reconfiguration of preconfigured Gi3 ingress policies of SP1R7 and SP1R8 (below config shows only SP1R7, please repeat the configuration on SP1R8 for completeness):

```
SP1R7#conf t
SP1R7(config)#policy-map pm-qos-customer-ingress
SP1R7(config-pmap)#class cm-qos-svc1
SP1R7(config-pmap-c)#set mpls experimental imposition 4
SP1R7(config-pmap-c)#class cm-qos-svc2
SP1R7(config-pmap-c)#set mpls experimental imposition 4
SP1R7(config-pmap-c)#class cm-qos-svc3
SP1R7(config-pmap-c)#set mpls experimental imposition 0
```

The policy should now remark our ingress traffic with proper MPLS EXP/TC values:

```
SP1R7#sh policy-map interface gigabitEthernet 3
 GigabitEthernet3
  Service-policy input: pm-qos-customer-ingress
    Class-map: cm-qos-svc1 (match-all)
      8 packets, 1200 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name acl-qos-svc1
      QoS Set
        mpls experimental imposition 4
          Marker statistics: Disabled

    Class-map: cm-qos-svc2 (match-all)
      8 packets, 1200 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name acl-qos-svc2
      QoS Set
        mpls experimental imposition 4
          Marker statistics: Disabled

    Class-map: cm-qos-svc3 (match-all)
      8 packets, 1200 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name acl-qos-svc3
      QoS Set
        mpls experimental imposition 0
          Marker statistics: Disabled

    Class-map: class-default (match-any)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      17 packets, 1294 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

Let's check if something did change on MPLS egress router – SP1R8 – we should start seeing MPLS EXP/TC values of 0 and 4 which were absent previously in the counters:

```
SP1R8#sh policy-map interface gigabitEthernet 2 | i Class|packets,
    Class-map: mpls-exp0 (match-all)
      4 packets, 616 bytes
    Class-map: mpls-exp1 (match-all)
      93 packets, 14322 bytes
    Class-map: mpls-exp2 (match-all)
      93 packets, 14322 bytes
    Class-map: mpls-exp3 (match-all)
      93 packets, 14322 bytes
    Class-map: mpls-exp4 (match-all)
      8 packets, 1232 bytes    ! notice, we're receiving twice as much traffic for this
                               ! class, and this is consistent with our marking scheme –
                               ! two probes are remarked with EXP4 while only one is with
                               ! EXP0
    Class-map: mpls-exp5 (match-all)
      0 packets, 0 bytes
    Class-map: mpls-exp6 (match-all)
      222 packets, 17103 bytes
    Class-map: mpls-exp7 (match-all)
      0 packets, 0 bytes
    Class-map: class-default (match-any)
      1533 packets, 140961 bytes
```

Indeed, only mpls-exp0 and mpls-exp4 class-maps are incrementing, as we can see after waiting for a moment:

```
SP1R8#sh policy-map interface gigabitEthernet 2 | i Class|packets,
    Class-map: mpls-exp0 (match-all)
      8 packets, 1232 bytes
    Class-map: mpls-exp1 (match-all)
      93 packets, 14322 bytes
    Class-map: mpls-exp2 (match-all)
      93 packets, 14322 bytes
    Class-map: mpls-exp3 (match-all)
      93 packets, 14322 bytes
    Class-map: mpls-exp4 (match-all)
      16 packets, 2464 bytes
    Class-map: mpls-exp5 (match-all)
      0 packets, 0 bytes
    Class-map: mpls-exp6 (match-all)
      230 packets, 17721 bytes
    Class-map: mpls-exp7 (match-all)
      0 packets, 0 bytes
    Class-map: class-default (match-any)
      1570 packets, 144304 bytes
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.1.3: Queueing on SP1R7 and SP1R8

As we're popping last label on PE router egressing towards IP-only network (towards customer), we have to make sure in pipe model we queue traffic towards IP-only network based on SP MPLS EXP/TC marking.

One of the ways is to map incoming, labelled traffic to QoS groups (internal and local to the router), and then queue the egressing traffic based on those groups, not on the IP QoS fields. We can achieve this by properly mapping the traffic. As a SP we again need to make some assumptions for our scenario. Let's say, as a SP we have following policy/contract with Customers:

- there's priority class, for up to 1Mbps, that we transport as EXP/TC 4
- there's best effort class, for up to 0.5Mbps, that we transport as EXP/TC of 0



Please note, that to properly reflect the sub-rate Ethernet service in this case, the shaper would need to be hierarchical – first level should shape down to some agreed value (like 3Mbps) an then class definition should follow and 2nd level. Otherwise, the queuing may not even kick in given that we're dealing with 1GE interface. However, this is for illustration and simplicity only.

Our egress policy on PE will need to reflect that:

```
SP1R8#conf t
SP1R8(config)#policy-map pm-mpls-core
SP1R8(config-pmap)#class mpls-exp0
SP1R8(config-pmap-c)#set qos-group 0
SP1R8(config-pmap-c)#class mpls-exp4
SP1R8(config-pmap-c)#set qos-group 4
SP1R8(config-pmap-c)#exit
SP1R8(config-pmap)#exit
SP1R8(config)#class-map cm-qosg-0
SP1R8(config-cmap)#match qos-group 0
SP1R8(config-cmap)#class-map cm-qosg-4
SP1R8(config-cmap)#match qos-group 4
SP1R8(config-cmap)#exit
SP1R8(config)#policy-map pm-qos-customer-egress
SP1R8(config-pmap)#class cm-qosg-4
SP1R8(config-pmap-c)#priority 1000
SP1R8(config-pmap-c)#exit
SP1R8(config-pmap)#class cm-qosg-0
SP1R8(config-pmap-c)#bandwidth 500
SP1R8(config-pmap-c)#exit
SP1R8(config-pmap)#exit
SP1R8(config)#int gi3
SP1R8(config-if)#service-policy output pm-qos-customer-egress
```

The traffic towards CER6 should be now mapped to proper classes as per our specification:

```
SP1R8#sh policy-map interface gigabitEthernet 3 output
 GigabitEthernet3

  Service-policy output: pm-qos-customer-egress
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    queue stats for all priority classes:
      Queueing
      queue limit 512 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 4/600

    Class-map: cm-qosg-4 (match-all)
      4 packets, 600 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: qos-group 4
      Priority: 1000 kbps, burst bytes 25000, b/w exceed drops: 0


    Class-map: cm-qosg-0 (match-all)
      9 packets, 834 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: qos-group 0
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 9/834
      bandwidth 500 kbps

    […]
```

To verify, that the exercise was executed correctly, we can setup a empty ingress policy on CER6 to check IP markings on the traffic – they should be identical to the ones configured on CEL6 IP SLA:

```
CER6#conf t
CER6(config)#class-map cm-qos-p0
CER6(config-cmap)#match precedence 0
CER6(config-cmap)#class-map cm-qos-p1
CER6(config-cmap)#match precedence 1
CER6(config-cmap)#class-map cm-qos-p2
CER6(config-cmap)#match precedence 2
CER6(config-cmap)#class-map cm-qos-p3
CER6(config-cmap)#match precedence 3
CER6(config-cmap)#class-map cm-qos-p4
CER6(config-cmap)#match precedence 4
CER6(config-cmap)#class-map cm-qos-p5
CER6(config-cmap)#match precedence 5
CER6(config-cmap)#class-map cm-qos-p6
CER6(config-cmap)#match precedence 6
CER6(config-cmap)#class-map cm-qos-p7
CER6(config-cmap)#match precedence 7
CER6(config-cmap)#exit
CER6(config)#policy-map CHECK-QOS-INGRESS
CER6(config-pmap)#class cm-qos-p0
CER6(config-pmap-c)#class cm-qos-p1
CER6(config-pmap-c)#class cm-qos-p2
CER6(config-pmap-c)#class cm-qos-p3
CER6(config-pmap-c)#class cm-qos-p4
CER6(config-pmap-c)#class cm-qos-p5
CER6(config-pmap-c)#class cm-qos-p6
CER6(config-pmap-c)#class cm-qos-p7
CER6(config-pmap-c)#exit
CER6(config)#int gi3
CER6(config-if)#service-policy input CHECK-QOS-INGRESS
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

…and then just watch the counters incrementing – for IP Precedence 1, 2 and 3:

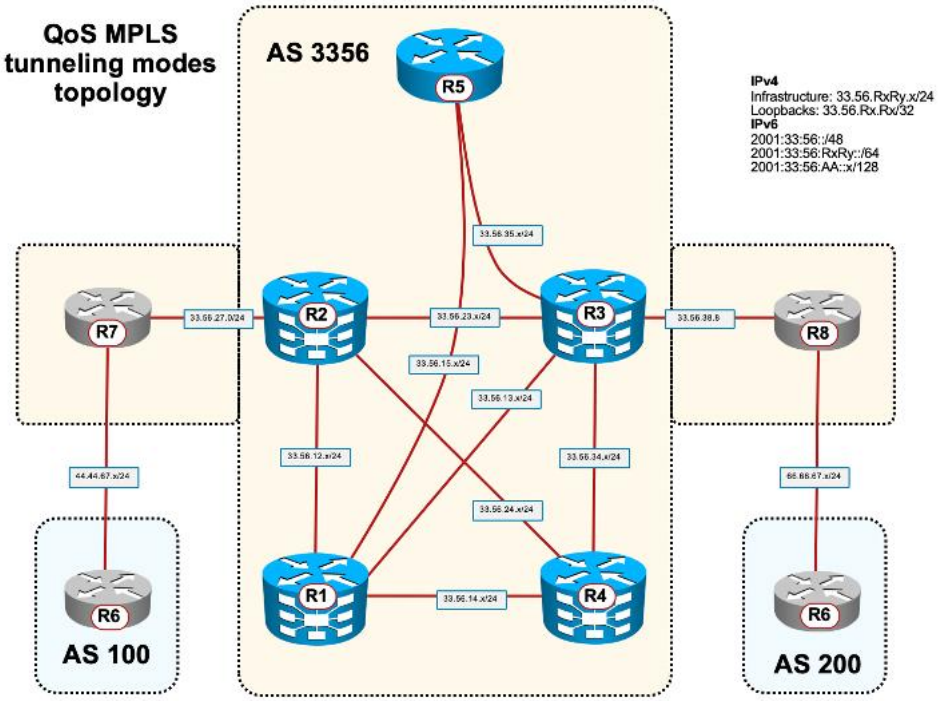
```
CER6#sh policy-map interface gigabitEthernet 3 | i Class|packets,
    Class-map: cm-qos-p0 (match-all)
      0 packets, 0 bytes
    Class-map: cm-qos-p1 (match-all)
      6 packets, 900 bytes
    Class-map: cm-qos-p2 (match-all)
      6 packets, 900 bytes
    Class-map: cm-qos-p3 (match-all)
      6 packets, 900 bytes
    Class-map: cm-qos-p4 (match-all)
      0 packets, 0 bytes
    Class-map: cm-qos-p5 (match-all)
      0 packets, 0 bytes
    Class-map: cm-qos-p6 (match-all)
      14 packets, 1029 bytes
    Class-map: cm-qos-p7 (match-all)
      1 packets, 118 bytes
    Class-map: class-default (match-any)
      0 packets, 0 bytes
```

Please repeat the configuration for other direction – for traffic flowing from CER8 to CER7 (by configuring proper policies on PE routers).

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5: Implementing short pipe MPLS QoS model

Building on the configuration already done in Task 4, we can now modify the config to implement short pipe model in our network:



Task 5.1: Configuring PE routers – SP1R7 and SP1R8

For short pipe, we need to change the way we queue traffic on egress to IP-only network from MPLS cloud. In pipe model, we were queueing on EXP values by mapping them to QoS groups.

In short pipe, we actually need to queue based on exposed IP QoS field of the traffic (so, original marking). The easiest way is just to remove QoS mapping or adjust existing one to match requirements.

On SP1R7 and SP1R8:

```
SP1R8#conf t
SP1R8(config)#int gi3
SP1R8(config-if)#no service-policy output pm-qos-customer-egress
```

And then let's recheck the counters on ingress of R6 routers (after clearing the counters):

```
CER6#clear counters gigabitEthernet 3
CER6#sh policy-map interface gigabitEthernet 3 | i Class|packets,
    Class-map: cm-qos-p0 (match-all)
      0 packets, 0 bytes
    Class-map: cm-qos-p1 (match-all)
      2 packets, 300 bytes
    Class-map: cm-qos-p2 (match-all)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      2 packets, 300 bytes
    Class-map: cm-qos-p3 (match-all)
      2 packets, 300 bytes
    Class-map: cm-qos-p4 (match-all)
      0 packets, 0 bytes
    Class-map: cm-qos-p5 (match-all)
      0 packets, 0 bytes
    Class-map: cm-qos-p6 (match-all)
      2 packets, 128 bytes
    Class-map: cm-qos-p7 (match-all)
      0 packets, 0 bytes
    Class-map: class-default (match-any)
      0 packets, 0 bytes
```

We can of course define some basic queueing on egress from SP1R8 towards AS200 (CER6), to work based on exposed IP QoS header:

```
SP1R8#conf t
SP1R8(config)#policy-map pm-qos-customer-egress
SP1R8(config-pmap)#no class cm-qosg-4
SP1R8(config-pmap)#no class cm-qosg-0
SP1R8(config-pmap)#class class-default
SP1R8(config-pmap-c)#random-detect precedence-based
SP1R8(config-pmap-c)#exit
SP1R8(config-pmap)#exit
SP1R8(config)#int gi3
SP1R8(config-if)#service-policy output pm-qos-customer-egress
```

And to verify the RED based on IP Precedence:

```
SP1R8#sh policy-map interface gigabitEthernet 3
 GigabitEthernet3

  Service-policy output: pm-qos-customer-egress

    Class-map: class-default (match-any)
      16 packets, 2094 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: any

      queue limit 416 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 16/2094
        Exp-weight-constant: 9 (1/512)
        Mean queue depth: 0 packets
        class  Transmitted    Random drop      Tail drop     Minimum        Maximum
Mark
               pkts/bytes     pkts/bytes       pkts/bytes    thresh         thresh
prob

        0          0/0           0/0              0/0              104         208
1/10
        1          4/600         0/0              0/0              117         208
1/10
        2          4/600         0/0              0/0              130         208
1/10
        3          4/600         0/0              0/0              143         208
1/10
        4          0/0           0/0              0/0              156         208
1/10
        5          0/0           0/0              0/0              169         208
1/10
        6          4/294         0/0              0/0              182         208
1/10
        7          0/0           0/0              0/0              195         208.
1/10
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 14: Segment Routing

Task 1: Segment Routing in OSPF domain of AS3356

This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch14.virl1.virl`
- **For VIRL²/CML²:** `sp.bootcamp.ch14.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch14.unl`

Task 1.1: Configuring Segment Routing for OSPF on IOS XR

Globally enabling SR on router:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#segment-routing
RP/0/0/CPU0:SP1R1(config-sr)#commit
RP/0/0/CPU0:SP1R1(config-sr)#
```

Enabling OSPF AS3356 instance to support SR forwarding:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#router ospf AS3356
RP/0/0/CPU0:SP1R1(config-ospf)#area 0
RP/0/0/CPU0:SP1R1(config-ospf-ar)#segment-routing mpls
RP/0/0/CPU0:SP1R1(config-ospf-ar)#segment-routing forwarding mpls
```

Let's check SR default global block allocation:

```
RP/0/0/CPU0:SP1R1#sh mpls label table detail | i SR
  (Lbl-blk SRLB, vers:0, (start_label=15000, size=1000, app_notify=0)
   ! starting from 6.3.1 release, this block is reserved by default for local mappings
  (Lbl-blk SRGB, vers:0, (start_label=16000, size=8000)
  (SR Adj Segment IPv4, vers:0, index=0, type=2, intf=Gi0/0/0/0, nh=33.56.23.3)
  (SR Adj Segment IPv4, vers:0, index=0, type=2, intf=Gi0/0/0/2, nh=33.56.12.1)
  (SR Adj Segment IPv4, vers:0, index=0, type=2, intf=Gi0/0/0/4, nh=33.56.24.4)
```

Please repeat the configuration on R2, R3 and R4.

Now, let's allocate prefix SIDs to each router within AS3356 by using the router number as index number (vs absolute values, which is still an option).

On SP1R1:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#router ospf AS3356
RP/0/0/CPU0:SP1R1(config-ospf)#area 0
RP/0/0/CPU0:SP1R1(config-ospf-ar)#interface loopback 0
RP/0/0/CPU0:SP1R1(config-ospf-ar-if)#prefix-sid index 1
RP/0/0/CPU0:SP1R1(config-ospf-ar-if)#commit
RP/0/0/CPU0:SP1R1(config-ospf-ar-if)#
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

..and on SP1R2, SP1R3 and up to SP1R4:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#router ospf AS3356
RP/0/0/CPU0:SP1R4(config-ospf)#area 0
RP/0/0/CPU0:SP1R4(config-ospf-ar)#interface loopback 0
RP/0/0/CPU0:SP1R4(config-ospf-ar-if)#prefix-sid index X ! where X equals router number
RP/0/0/CPU0:SP1R4(config-ospf-ar-if)#
```

If you mess up somewhere with assigning interfaces, IOS XR will fall back of not enabling the erroneous mapping if it already exists. Like for example here:

```
RP/0/0/CPU0: ospf[1028]: %ROUTING-OSPF-3-SR_SID_CONFLICT : Prefix SID index 1 assigned
to 33.56.2.2/32 also used by 33.56.1.1/32 - ignoring SID until conflict resolved
```

As we don't yet enabled SR on R5, on R3 we can see three Node Prefixes and multiple SR Adjacencies:

```
RP/0/0/CPU0:SP1R4#c
16001  Pop          SR Pfx (idx 1)     Gi0/0/0/0    33.56.14.1      6400
16002  Pop          SR Pfx (idx 2)     Gi0/0/0/5    33.56.24.2      15298
16003  Pop          SR Pfx (idx 3)     Gi0/0/0/2    33.56.34.3      5500
104037 Pop          SR Adj (idx 0)     Gi0/0/0/0    33.56.14.1      0
104038 Pop          SR Adj (idx 0)     Gi0/0/0/0    33.56.14.1      0
104040 Pop          SR Adj (idx 0)     Gi0/0/0/2    33.56.34.3      0
104041 Pop          SR Adj (idx 0)     Gi0/0/0/2    33.56.34.3      0
104042 Pop          SR Adj (idx 0)     Gi0/0/0/5    33.56.24.2      0
104043 Pop          SR Adj (idx 0)     Gi0/0/0/5    33.56.24.2      0
```

Same view on SP1R1:

```
RP/0/0/CPU0:SP1R1#sh mpls forwarding  | i SR
16002  Pop          SR Pfx (idx 2)     tt5000       33.56.2.2       0
16003  Pop          SR Pfx (idx 3)     tt5001       33.56.3.3       0
16004  Pop          SR Pfx (idx 4)     tt5002       33.56.4.4       0
101011 Pop          SR Adj (idx 0)     Gi0/0/0/0    33.56.14.4      0
101012 Pop          SR Adj (idx 0)     Gi0/0/0/0    33.56.14.4      0
101027 Pop          SR Adj (idx 0)     Gi0/0/0/3    33.56.15.5      0
101028 Pop          SR Adj (idx 0)     Gi0/0/0/3    33.56.15.5      0
101029 Pop          SR Adj (idx 0)     Gi0/0/0/5    33.56.13.3      0
101030 Pop          SR Adj (idx 0)     Gi0/0/0/5    33.56.13.3      0
```

Forwarding is still based on LDP labels:

```
RP/0/0/CPU0:SP1R4#sh mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16001  Pop         SR Pfx (idx 1)     Gi0/0/0/0    33.56.14.1      0
16002  Pop         SR Pfx (idx 2)     Gi0/0/0/5    33.56.24.2      0
16003  Pop         SR Pfx (idx 3)     Gi0/0/0/2    33.56.34.3      0
104000 Exp-Null-v4 33.56.2.2/32       Gi0/0/0/5    33.56.24.2      154
104001 Exp-Null-v4 33.56.1.1/32       Gi0/0/0/0    33.56.14.1      100

[…]

RP/0/0/CPU0:SP1R4#ping 33.56.1.1 size 100 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 33.56.1.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/2/9 ms

RP/0/0/CPU0:SP1R4#sh mpls forwarding
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
16001  Pop         SR Pfx (idx 1)    Gi0/0/0/0    33.56.14.1      0
16002  Pop         SR Pfx (idx 2)    Gi0/0/0/5    33.56.24.2      0
16003  Pop         SR Pfx (idx 3)    Gi0/0/0/2    33.56.34.3      0
104000 Exp-Null-v4 33.56.2.2/32      Gi0/0/0/5    33.56.24.2      540
104001 Exp-Null-v4 33.56.1.1/32      Gi0/0/0/0    33.56.14.1      10600
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: Configuring Segment Routing for OSPF on IOS-XE

Now, let's enable SR on R5 – IOS-XE device:

```
SP1-R5-RR#conf t
SP1-R5-RR(config)#segment-routing mpls
SP1-R5-RR(config)#router ospf 100
SP1-R5-RR(config-router)#segment-routing mpls
```

Notice, we haven't yet enabled Prefix SID for R5, so it's not yet advertised in the AS3356 (check on R1):

```
RP/0/0/CPU0:SP1R1#sh mpls forwarding | i SR
16002  Pop          SR Pfx (idx 2)     Gi0/0/0/2    33.56.12.2      0
16003  Pop          SR Pfx (idx 3)     Gi0/0/0/5    33.56.13.3      0
16004  Pop          SR Pfx (idx 4)     Gi0/0/0/0    33.56.14.4      0
! no 16005 label from SR
101009 Pop          SR Adj (idx 0)     Gi0/0/0/0    33.56.14.4      0
101010 Pop          SR Adj (idx 0)     Gi0/0/0/0    33.56.14.4      0
101011 Pop          SR Adj (idx 0)     Gi0/0/0/2    33.56.12.2      0
101012 Pop          SR Adj (idx 0)     Gi0/0/0/2    33.56.12.2      0
101013 Pop          SR Adj (idx 0)     Gi0/0/0/3    33.56.15.5      0
101014 Pop          SR Adj (idx 0)     Gi0/0/0/3    33.56.15.5      0
101025 Pop          SR Adj (idx 0)     Gi0/0/0/5    33.56.13.3      0
101026 Pop          SR Adj (idx 0)     Gi0/0/0/5    33.56.13.3      0
```

We need to enable mapping of SIDs:

```
SP1-R5-RR#conf t
SP1-R5-RR(config)#segment-routing mpls
SP1-R5-RR(config-srmpls)#connected-prefix-sid-map
SP1-R5-RR(config-srmpls-conn)#address-family ipv4
SP1-R5-RR(config-srmpls-conn-af)#33.56.5.5/32 index 5
SP1-R5-RR(config-srmpls-conn-af)#^Z
```

Label is flooded in AS3356 immediately:

```
RP/0/0/CPU0:SP1R1#sh mpls forwarding  | i SR
16002  Pop          SR Pfx (idx 2)     Gi0/0/0/2    33.56.12.2      0
16003  Pop          SR Pfx (idx 3)     Gi0/0/0/5    33.56.13.3      0
16004  Pop          SR Pfx (idx 4)     Gi0/0/0/0    33.56.14.4      0
16005  Pop          SR Pfx (idx 5)     Gi0/0/0/3    33.56.15.5      0
101013 Pop          SR Adj (idx 0)     Gi0/0/0/0    33.56.14.4      0
101014 Pop          SR Adj (idx 0)     Gi0/0/0/2    33.56.12.2      0
101015 Pop          SR Adj (idx 0)     Gi0/0/0/3    33.56.15.5      0
101016 Pop          SR Adj (idx 0)     Gi0/0/0/5    33.56.13.3      0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3: Verification

When executed correctly, all OSPF nodes should see Prefix-SID and Adjacency-SIDs flooded through the domain. For example on R1:

```
RP/0/0/CPU0:SP1R1#sh ospf sid-database
SID Database for ospf AS3356 with ID 33.56.1.1

SID          Prefix/Mask
--------     ------------------
1            33.56.1.1/32             (L) ! L signifies local mapping
2            33.56.2.2/32
3            33.56.3.3/32
4            33.56.4.4/32
5            33.56.5.5/32
```

…and on IOS-XE:

```
SP1-R5-RR#sh ip ospf segment-routing sid-database

            OSPF Router with ID (33.56.5.5) (Process ID 100)

OSPF Segment Routing SIDs

Flags: L - local, N - label not programmed,
       M - mapping-server

  SID               Prefix/Mask              Adv-Rtr-Id            Area-Id
--------          ------------------       ------------------     -------------
1                 33.56.1.1/32             33.56.1.1               0
2                 33.56.2.2/32             33.56.2.2               0
3                 33.56.3.3/32             33.56.3.3               0
4                 33.56.4.4/32             33.56.4.4               0
5           (L)   33.56.5.5/32             33.56.5.5               0
             ! 'L' has same meaning as with IOS-XR - it's local mapping
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4: Migrating AS3356 from LDP to SR

Task 1.4.1: Making SR preferred over LDP

On all nodes within AS3356 add following configuration entry to OSPF AS3356 process:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#router ospf AS3356
RP/0/0/CPU0:SP1R1(config-ospf)#segment-routing sr-prefer
RP/0/0/CPU0:SP1R1(config-ospf)#commit label SRPREFER ! this will save your time
```

For IOS-XE nodes the preference is specified in segment-routing global config part:

```
SP1-R5-RR#conf t
SP1-R5-RR(config)#segment-routing mpls
SP1-R5-RR(config-srmpls)#set-attributes
SP1-R5-RR(config-srmpls-attr)#address-family ipv4
SP1-R5-RR(config-srmpls-attr-af)#sr-label-preferred
SP1-R5-RR(config-srmpls-attr-af)#^Z
```

Task 1.4.2: Verification

We can verify, that creating traffic to the index 1 (which is SP1R1) from SP1R4 creates traffic on MPLS forwarding data plane:

```
RP/0/0/CPU0:SP1R4#sh mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16001  Pop         SR Pfx (idx 1)     Gi0/0/0/0    33.56.14.1      100
16002  Pop         SR Pfx (idx 2)     Gi0/0/0/5    33.56.24.2      146
16003  Pop         SR Pfx (idx 3)     Gi0/0/0/2    33.56.34.3      0
104000 Exp-Null-v4 33.56.2.2/32       Gi0/0/0/5    33.56.24.2      44
104001 Exp-Null-v4 33.56.1.1/32       Gi0/0/0/0    33.56.14.1      44

[…]

RP/0/0/CPU0:SP1R4#ping 33.56.1.1 size 100 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 33.56.1.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/2/9 ms

RP/0/0/CPU0:SP1R4#sh mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16001  Pop         SR Pfx (idx 1)     Gi0/0/0/0    33.56.14.1      10600
16002  Pop         SR Pfx (idx 2)     Gi0/0/0/5    33.56.24.2      392
16003  Pop         SR Pfx (idx 3)     Gi0/0/0/2    33.56.34.3      100
104000 Exp-Null-v4 33.56.2.2/32       Gi0/0/0/5    33.56.24.2      44
104001 Exp-Null-v4 33.56.1.1/32       Gi0/0/0/0    33.56.14.1      44

[…]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4.3: Turning off LDP in AS3356

Turning off MPLS LDP completely, and relying on SR to forward traffic:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#no mpls ldp
RP/0/0/CPU0:SP1R4(config)#commit label NOLDP
RP/0/0/CPU0:SP1R4(config)#
```

Note that currently, with IOS-XE you can't turn off LDP completely. Given SR is still preferred for FIB installation, you'll use SR provided entries anyway.

```
RP/0/0/CPU0:SP1R4#sh mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16001  Pop         SR Pfx (idx 1)     Gi0/0/0/0    33.56.14.1      12500
16002  Pop         SR Pfx (idx 2)     Gi0/0/0/5    33.56.24.2      5322
16003  Pop         SR Pfx (idx 3)     Gi0/0/0/2    33.56.34.3      2000

RP/0/0/CPU0:SP1R4#ping 33.56.1.1 size 100 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 33.56.1.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/2/9 ms

RP/0/0/CPU0:SP1R4#sh mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16001  Pop         SR Pfx (idx 1)     Gi0/0/0/0    33.56.14.1      23000
16002  Pop         SR Pfx (idx 2)     Gi0/0/0/5    33.56.24.2      5926
16003  Pop         SR Pfx (idx 3)     Gi0/0/0/2    33.56.34.3      2200

[…]

RP/0/0/CPU0:SP1R4#ping 33.56.3.3 size 100 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 33.56.3.3, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/2/9 ms

RP/0/0/CPU0:SP1R4#sh mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16001  Pop         SR Pfx (idx 1)     Gi0/0/0/0    33.56.14.1      23158
16002  Pop         SR Pfx (idx 2)     Gi0/0/0/5    33.56.24.2      6150
16003  Pop         SR Pfx (idx 3)     Gi0/0/0/2    33.56.34.3      12700
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Segment Routing in IS-IS domain of AS1239

Task 2.1: Configuring Segment Routing for IS-IS on IOS XR

AS1239 is running IS-IS as IGP. Let's configure the routers just as we did with AS3356. Your IOS XR config on all nodes (R1-R4) should look like this after configuration:

```
RP/0/0/CPU0:SP2R1#sh running-config segment-routing
segment-routing
!
RP/0/0/CPU0:SP2R1#sh running-config router isis AS1239
router isis AS1239
 is-type level-2-only
 net 1239.0000.0000.0001.00
 address-family ipv4 unicast
  metric-style wide
  segment-routing mpls
  mpls ldp auto-config
 !
interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 101 ! we use offset of 100 for AS1239 – so 1 becomes 101
  !
  address-family ipv6 unicast
  !
 !
```

Task 2.2: Configuring Segment Routing for IS-IS on IOS-XE

IOS-XE node – R5 – should be configured as follows:

```
!
segment-routing mpls
 connected-prefix-sid-map
  address-family ipv4
   12.39.5.5/32 index 105 range 1
 exit-address-family
!
router isis AS1239
 net 12.3900.0000.0000.0500
 metric-style wide
 segment-routing mpls
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.3: Verification

When executed correctly, all IS-IS nodes see Prefix-SID and Adjacency-SIDs flooded through the domain. For example on R2:

```
RP/0/0/CPU0:SP2R2#show isis database level 2 verbose | i Prefix-SID
    Prefix-SID Index: 101, Algorithm:0, R:0 N:1 P:0 E:0 V:0 L:0
    Prefix-SID Index: 102, Algorithm:0, R:0 N:1 P:0 E:0 V:0 L:0
    Prefix-SID Index: 103, Algorithm:0, R:0 N:1 P:0 E:0 V:0 L:0
    Prefix-SID Index: 104, Algorithm:0, R:0 N:1 P:0 E:0 V:0 L:0
    Prefix-SID Index: 105, Algorithm:0, R:0 N:1 P:0 E:0 V:0 L:0
```

On R5:

```
SP2-R5-RR#show isis database level-2 verbose | i Prefix-SID
    Prefix-SID Index: 101, Algorithm:SPF, R:0 N:1 P:0 E:0 V:0 L:0
    Prefix-SID Index: 102, Algorithm:SPF, R:0 N:1 P:0 E:0 V:0 L:0
    Prefix-SID Index: 103, Algorithm:SPF, R:0 N:1 P:0 E:0 V:0 L:0
    Prefix-SID Index: 104, Algorithm:SPF, R:0 N:1 P:0 E:0 V:0 L:0
    Prefix-SID Index: 105, Algorithm:SPF, R:0 N:1 P:0 E:0 V:0 L:0
```

And of course, label mappings are present in the LFIB – on R1:

```
RP/0/0/CPU0:SP2R1#sh mpls forwarding | i SR
16102  Pop          SR Pfx (idx 102)   Gi0/0/0/2    12.39.12.2      0
16103  Pop          SR Pfx (idx 103)   Gi0/0/0/4    12.39.13.3      0
16104  Pop          SR Pfx (idx 104)   Gi0/0/0/0    12.39.14.4      0
16105  Pop          SR Pfx (idx 105)   Gi0/0/0/3    12.39.15.5      0
123110 Pop          SR Adj (idx 1)     Gi0/0/0/2    12.39.12.2      0
123111 Pop          SR Adj (idx 3)     Gi0/0/0/2    12.39.12.2      0
123112 Pop          SR Adj (idx 1)     Gi0/0/0/0    12.39.14.4      0
123113 Pop          SR Adj (idx 3)     Gi0/0/0/0    12.39.14.4      0
123114 Pop          SR Adj (idx 1)     Gi0/0/0/4    12.39.13.3      0
123115 Pop          SR Adj (idx 3)     Gi0/0/0/4    12.39.13.3      0
123116 Pop          SR Adj (idx 1)     Gi0/0/0/3    12.39.15.5      0
123117 Pop          SR Adj (idx 3)     Gi0/0/0/3    12.39.15.5      0
```

..and on R5 (for example):

```
SP2-R5-RR#sh mpls forwarding-table | i 1610
16101      Pop Label  12.39.1.1/32     0             Gi3        12.39.15.1
16102      16102      12.39.2.2/32     0             Gi3        12.39.15.1
           16102      12.39.2.2/32     0             Gi2        12.39.35.3
16103      Pop Label  12.39.3.3/32     0             Gi2        12.39.35.3
16104      16104      12.39.4.4/32     0             Gi3        12.39.15.1
           16104      12.39.4.4/32     0             Gi2        12.39.35.3
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.4: Migrating AS1239 from LDP to SRs

Task 2.4.1: Making SR preferred over LDP

On all nodes within AS 1239 add following configuration entry to IS-IS process:

```
RP/0/0/CPU0:SP2R1#conf t
RP/0/0/CPU0:SP2R1(sconfig)#router isis AS1239
RP/0/0/CPU0:SP2R1(config-isis)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R1(config-isis-af)#segment-routing mpls sr-prefer
RP/0/0/CPU0:SP2R1(config-isis-af)#commit label SRPREFER ! this will save your time
```

For IOS-XE nodes the preference is specified in segment-routing global config part:

```
SP2-R5-RR#conf t
SP2-R5-RR(config)#segment-routing mpls
SP2-R5-RR(config-srmpls)#set-attributes
SP2-R5-RR(config-srmpls-attr)#address-family ipv4
SP2-R5-RR(config-srmpls-attr-af)#sr-label-preferred
SP2-R5-RR(config-srmpls-attr-af)#^Z
```

Task 2.4.2: Verification

We can verify, that creating traffic to the index 1 (which is SP1R1) from SP1R4 creates traffic on MPLS forwarding data plane:

```
RP/0/0/CPU0:SP2R4#sh mpls forwarding | i 16101
16101  Pop          SR Pfx (idx 101)   Gi0/0/0/0    12.39.14.1      5109
RP/0/0/CPU0:SP2R4#ping 12.39.1.1 so lo0 size 100 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 12.39.1.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/2/9 ms
RP/0/0/CPU0:SP2R4#sh mpls forwarding | i 16101
16101  Pop          SR Pfx (idx 101)   Gi0/0/0/0    12.39.14.1      15549
```

Task 2.4.3: Turning off LDP in AS 1239

Turning off MPLS LDP completely, and relying on SR to forward traffic:

```
RP/0/0/CPU0:SP2R4#conf t
RP/0/0/CPU0:SP2R4(config)#no mpls ldp
RP/0/0/CPU0:SP2R4(config)#commit label NOLDP
RP/0/0/CPU0:SP2R4(config)#
```

Note that currently, with IOS-XE you can't turn off LDP completely. Given SR is still preferred for FIB installation, you'll use SR provided entries anyway.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3: Segment Routing TE in IOS XR pre-6.3

Task 3.1: Simple SR TE with MPLS data plane with OSPF

Configuring TE requires enabling of MPLS TE capabilities, just like with plain old MPLS. Please repeat this (adjusting for interfaces specific to a router) in all routers in AS3356:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#router ospf AS3356
RP/0/0/CPU0:SP1R1(config-ospf)#area 0
RP/0/0/CPU0:SP1R1(config-ospf-ar)#mpls traffic-eng
RP/0/0/CPU0:SP1R1(config-ospf-ar)#exit
RP/0/0/CPU0:SP1R1(config-ospf)#exit
RP/0/0/CPU0:SP1R1(config)#mpls traffic-eng
RP/0/0/CPU0:SP1R1(config-mpls-te)#int gi0/0/0/0
RP/0/0/CPU0:SP1R1(config-mpls-te-if)#int gi0/0/0/2
RP/0/0/CPU0:SP1R1(config-mpls-te-if)#int gi0/0/0/3
RP/0/0/CPU0:SP1R1(config-mpls-te-if)#int gi0/0/0/5
```

…and on R5, given its IOS-XE:

```
SP1-R5-RR#conf t
SP1-R5-RR(config)#mpls traffic-eng tunnels
SP1-R5-RR(config)#int gi2
SP1-R5-RR(config-if)#mpls ip
SP1-R5-RR(config-if)#mpls traffic-eng tunnels
SP1-R5-RR(config-if)#int gi3
SP1-R5-RR(config-if)#mpls ip
SP1-R5-RR(config-if)#mpls traffic-eng tunnels
SP1-R5-RR(config)#router ospf 100
SP1-R5-RR(config-router)#mpls traffic-eng router-id lo0
SP1-R5-RR(config-router)#mpls traffic-eng area 0
```

Notice, we're no longer configuring RSVP - it's not needed anymore.

Before we continue, let's check all nodes are in MPLS TE database:

```
RP/0/0/CPU0:SP1R1#sh mpls traffic-eng topology | i MPLS TE Id
IGP Id: 33.56.1.1, MPLS TE Id: 33.56.1.1 Router Node  (OSPF AS3356 area 0)
IGP Id: 33.56.2.2, MPLS TE Id: 33.56.2.2 Router Node  (OSPF AS3356 area 0)
IGP Id: 33.56.3.3, MPLS TE Id: 33.56.3.3 Router Node  (OSPF AS3356 area 0)
IGP Id: 33.56.4.4, MPLS TE Id: 33.56.4.4 Router Node  (OSPF AS3356 area 0)
IGP Id: 33.56.5.5, MPLS TE Id: 33.56.5.5 Router Node  (OSPF AS3356 area 0)
```



In the older versions of IOS XR (pre 6.4), SR TE was done exactly like typical MPLS TE configuration - the only difference was that 'segment' routing was added to path option specification under TE interface.

Note that this config won't work in IOS XR releases newer than 6.4.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Let's configure SR TE tunnel going from R4 to R2, R3, R1 and then ending up at R5. For that, we need explicit-path. Notice however, we're using SID labels, not IPv4 addresses for next-hops:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#explicit-path name R42315
RP/0/0/CPU0:SP1R4(config-expl-path)#index 10 next-label 16002
RP/0/0/CPU0:SP1R4(config-expl-path)#index 20 next-label 16003
RP/0/0/CPU0:SP1R4(config-expl-path)#index 30 next-label 16001
RP/0/0/CPU0:SP1R4(config-expl-path)#index 40 next-label 16005
RP/0/0/CPU0:SP1R4(config-expl-path)#exit
```

Now, for the tunnel definition:

```
RP/0/0/CPU0:SP1R4(config)#inte tunnel-te 42315
RP/0/0/CPU0:SP1R4(config-if)#ipv4 unnumbered loopback 0
RP/0/0/CPU0:SP1R4(config-if)#destination 33.56.5.5
RP/0/0/CPU0:SP1R4(config-if)#path-option 10 explicit name R42315 segment-routing
RP/0/0/CPU0:SP1R4(config-if)#autoroute announce
```

Tunnel should be set up immediately using SR signalling:

```
RP/0/0/CPU0:SP1R4#sh mpls traffic-eng tunnels 42315
Name: tunnel-te42315  Destination: 33.56.5.5  Ifhandle:0xf0
  Signalled-Name: SP1R4_t42315
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
    path option 10, (Segment-Routing) type explicit R42315 (Basis for Setup)
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Mon Jan 01 22:35:39 2017 (00:00:08 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
      Protection: any (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Disabled
  History:
    Tunnel has been up for: 00:00:07 (since Mon Jan 01 22:35:40 UTC 2017)
    Current LSP:
      Uptime: 00:00:07 (since Mon Jan 01 22:35:40 UTC 2017)

  Segment-Routing Path Info (OSPF AS3356 area 0)
    Segment0[Node]: 33.56.2.2, Label: 16002
    Segment1[Node]: 33.56.3.3, Label: 16003
    Segment2[Node]: 33.56.1.1, Label: 16001
    Segment3[Node]: 33.56.5.5, Label: 16005
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

…and Autoroute announce should install the routing entry to tunnel destination – 33.56.5.5:

```
RP/0/0/CPU0:SP1R4#sh route 33.56.5.5
Routing entry for 33.56.5.5/32
  Known via "ospf AS3356", distance 110, metric 3, labelled SR, type intra area
  Installed Jan 01 22:37:22.540 for 00:00:23
  Routing Descriptor Blocks
    33.56.5.5, from 33.56.5.5, via tunnel-te42315
      Route metric is 3
  No advertising protos.
```

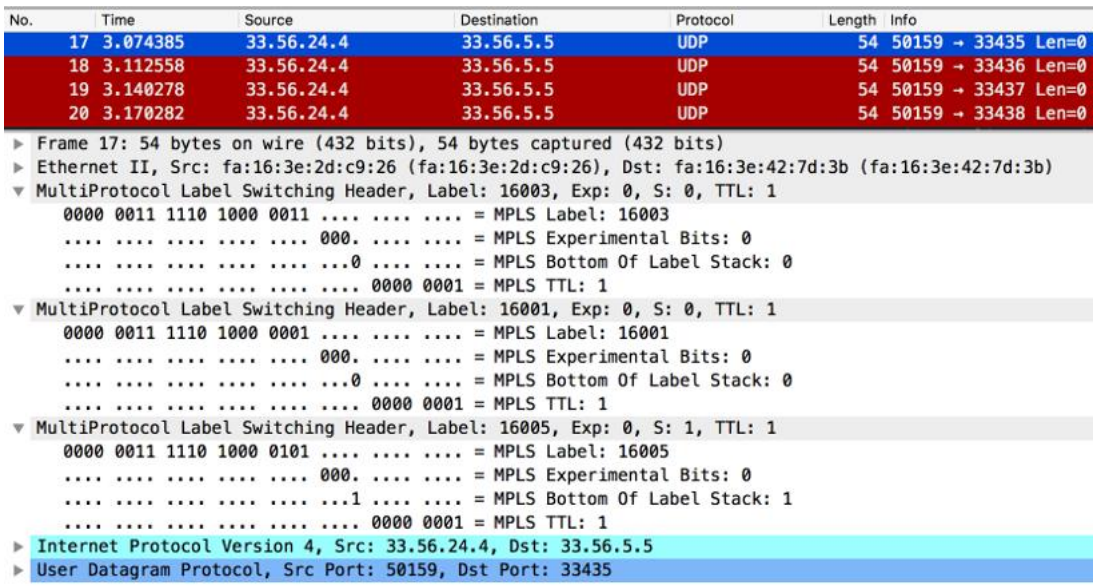
The tunnel should also immediately pass traffic:

```
RP/0/0/CPU0:SP1R4#sh mpls forwarding tunnels
Tunnel         Outgoing     Outgoing      Next Hop        Bytes
Name           Label        Interface                     Switched
-------------- ------------ ------------ --------------- ------------
tt42315      (SR) 16003       Gi0/0/0/5    33.56.24.2      6833
```

Now, let's check the label stack used by SR TE – notice it's truly a label stack compared to normal MPLS TE:

```
RP/0/0/CPU0:SP1R4#traceroute 33.56.5.5
Type escape sequence to abort.
Tracing the route to 33.56.5.5
 1  33.56.24.2 [MPLS: Labels 16003/16001/16005 Exp 0] 9 msec  9 msec  9 msec
 2  33.56.23.3 [MPLS: Labels 16001/16005 Exp 0] 9 msec  9 msec  9 msec
 3  33.56.13.1 [MPLS: Label 16005 Exp 0] 9 msec  9 msec  9 msec
 4  33.56.15.5 9 msec  *  9 msec
```

Label stack can be verified in packet capture – notice three labels in the traceroute UDP packet:



© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2: Simple SR TE with MPLS data plane with IS-IS

Configuring TE requires enabling of MPLS TE capabilities, just like with plain old MPLS. Please repeat this (adjusting for interfaces specific to a router) in all routers in AS1239:

```
RP/0/0/CPU0:SP2R1#conf t
RP/0/0/CPU0:SP2R1(config)#router isis AS1239
RP/0/0/CPU0:SP2R1(config-isis)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R1(config-isis-af)#mpls traffic-eng level-2-only
RP/0/0/CPU0:SP2R1(config-isis-af)#mpls traffic-eng router-id lo0
RP/0/0/CPU0:SP2R1(config-isis-af)#mpls traffic-eng
RP/0/0/CPU0:SP2R1(config-mpls-te)#int gi0/0/0/0
RP/0/0/CPU0:SP2R1(config-mpls-te-if)#int gi0/0/0/2
RP/0/0/CPU0:SP2R1(config-mpls-te-if)#int gi0/0/0/3
RP/0/0/CPU0:SP2R1(config-mpls-te-if)#int gi0/0/0/4
RP/0/0/CPU0:SP2R1(config-mpls-te-if)#exit
RP/0/0/CPU0:SP2R1(config-mpls-te)#exit
RP/0/0/CPU0:SP2R1(config)#commit label SR-MPLS-TE
```

…and on R5, given its IOS-XE:

```
SP2R5-RR#conf t
SP2R5-RR(config)#router isis AS1239
SP2R5-RR(config-router)#mpls traffic-eng level-2
SP2R5-RR(config-router)#mpls traffic-eng router-id lo0
```

Notice, we're no longer configuring RSVP - it's not needed anymore.

Before we continue, let's check all nodes are in MPLS TE database:

```
SP2R5-RR#sh mpls traffic-eng topology | i MPLS TE Id
IGP Id: 0000.0000.0001.00, MPLS TE Id:12.39.1.1 Router Node  (isis  level-2)
IGP Id: 0000.0000.0002.00, MPLS TE Id:12.39.2.2 Router Node  (isis  level-2)
IGP Id: 0000.0000.0003.00, MPLS TE Id:12.39.3.3 Router Node  (isis  level-2)
IGP Id: 0000.0000.0004.00, MPLS TE Id:12.39.4.4 Router Node  (isis  level-2)
IGP Id: 0000.0000.0005.00, MPLS TE Id:12.39.5.5 Router Node  (isis  level-2)
```



In the older versions of IOS XR (pre 6.4), SR TE was done exactly like typical MPLS TE configuration - the only difference was that 'segment' routing was added to path option specification under TE interface.

Note that this config won't work in IOS XR releases newer than 6.4.

Let's configure SR TE tunnel going from R4 to R2, R3, R1 and then ending up at R5. For that, we need explicit-path. Notice however, we're using SID labels, not IPv4 addresses for next-hops:

```
RP/0/0/CPU0:SP2R4#conf t
RP/0/0/CPU0:SP2R4(config)#explicit-path name R42315
RP/0/0/CPU0:SP2R4(config-expl-path)#index 10 next-label 16102
RP/0/0/CPU0:SP2R4(config-expl-path)#index 20 next-label 16103
RP/0/0/CPU0:SP2R4(config-expl-path)#index 30 next-label 16101
RP/0/0/CPU0:SP2R4(config-expl-path)#index 40 next-label 16105
RP/0/0/CPU0:SP2R4(config-expl-path)#exit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Now, for the tunnel definition:

```
RP/0/0/CPU0:SP2R4(config)#inte tunnel-te 42315
RP/0/0/CPU0:SP2R4(config-if)#ipv4 unnumbered loopback 0
RP/0/0/CPU0:SP2R4(config-if)#destination 12.39.5.5
RP/0/0/CPU0:SP2R4(config-if)#path-option 10 explicit name R42315 segment-routing
RP/0/0/CPU0:SP2R4(config-if)#autoroute announce
RP/0/0/CPU0:SP2R4(config-if-tunte-aa)#commit
```

Tunnel should be set up immediately using SR signalling:

```
RP/0/0/CPU0:SP2R4#sh mpls traffic-eng tunnels 42315

Name: tunnel-te42315  Destination: 12.39.5.5  Ifhandle:0x70
  Signalled-Name: SP2R4_t42315
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
    path option 10, (Segment-Routing) type explicit R42315 (Basis for Setup)
 [...]
    SRLG discovery: Disabled
  History:
    Tunnel has been up for: 00:01:50 (since Mon Jan 01 10:57:43 UTC 2020)
    Current LSP:
      Uptime: 00:01:50 (since Mon Jan 01 10:57:43 UTC 2020)

  Segment-Routing Path Info (IS-IS AS1239 level-2)
    Segment0[Node]: 12.39.2.2, Label: 16102
    Segment1[Node]: 12.39.3.3, Label: 16103
    Segment2[Node]: 12.39.1.1, Label: 16101
    Segment3[Node]: 12.39.5.5, Label: 16105
```

…and Autoroute announce should install the routing entry to tunnel destination – 33.56.5.5:

```
RP/0/0/CPU0:SP2R4#sh route 12.39.5.5
Routing entry for 12.39.5.5/32
  Known via "isis AS1239", distance 115, metric 30, labeled SR, label redist non FIB,
type level-2
  Routing Descriptor Blocks
    12.39.5.5, from 12.39.5.5, via tunnel-te42315
      Route metric is 30
  No advertising protos.
```

The tunnel should also immediately pass traffic:

```
RP/0/0/CPU0:SP2R4#sh mpls forwarding tunnels
Tunnel                    Outgoing    Outgoing     Next Hop        Bytes
Name                      Label       Interface                    Switched
------------------------- ----------- ------------ --------------- ------------
tunnel-te42315        (SR) 16103       Gi0/0/0/5    12.39.24.2      799
```

Now, let's check the label stack used by SR TE – notice it's truly a label stack compared to normal MPLS TE:

```
RP/0/0/CPU0:SP2R4#traceroute 12.39.5.5 so lo0

Type escape sequence to abort.
Tracing the route to 12.39.5.5

 1  12.39.24.2 [MPLS: Labels 16103/16101/16105 Exp 0] 9 msec  9 msec  0 msec
 2  12.39.23.3 [MPLS: Labels 16101/16105 Exp 0] 9 msec  9 msec  9 msec
 3  12.39.13.1 [MPLS: Label 16105 Exp 0] 9 msec  9 msec  9 msec
 4  12.39.15.5 9 msec  *  9 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4: Segment Routing interworking

Task 4.1: Segment Routing Mapping Server feature

To execute this testing, we'll put SP2R4 into LDP-only domain:

- on SP2R4:
  - disable SR on SP2R4
  - enable LDP on SP2R4
  - shut down gi0/0/0/2 and gi0/0/0/5
- on SP2R1
  - shut down gi0/0/0/3 and gi0/0/0/4
  - on all other routers in AS 1239, make sure LDP is disabled

From SP2R4 traceroute to SP2R5 should look like this:

```
RP/0/0/CPU0:SP2R4#traceroute 12.39.5.5 source loopback 0

Type escape sequence to abort.
Tracing the route to 12.39.5.5

 1  12.39.14.1 0 msec  0 msec  0 msec
 2  12.39.12.2 [MPLS: Label 16105 Exp 0] 9 msec  9 msec  9 msec
 3  12.39.23.3 [MPLS: Label 16105 Exp 0] 9 msec  9 msec  9 msec
 4  12.39.35.5 9 msec  *  9 msec
```

Notice that we don’t have label for first hop, then there’s 16105 provided by SR and on last hop as PHP is in effect there’s again no label.

On the SP2R5 the output should show no labels - destination is not advertiesed currently in any label protocol:

```
SP2R5-RR#traceroute 12.39.4.4 source lo0
Type escape sequence to abort.
Tracing the route to 12.39.4.4
VRF info: (vrf in name/id, vrf out name/id)
  1 12.39.35.3 2 msec 2 msec 1 msec
  2 12.39.23.2 6 msec 6 msec 5 msec
  3 12.39.12.1 8 msec 7 msec 6 msec
  4 12.39.14.4 10 msec *  11 msec
```

Now, we'll configure SR Mapping Server on S2R3. Location of Mapping Server is irrelevant - it just has to be within common IGP domain, as the announcement will be flooded. Good practice dictates, that we should configure two, but as we're working in lab scenario, one Mapping Server will do for now:

```
RP/0/0/CPU0:SP2R3#conf t
RP/0/0/CPU0:SP2R3(config)#segment-routing
RP/0/0/CPU0:SP2R3(config-sr)#mapping-server
RP/0/0/CPU0:SP2R3(config-sr-ms)#prefix-sid-map
RP/0/0/CPU0:SP2R3(config-sr-ms-map)#address-family ipv4
RP/0/0/CPU0:SP2R3(config-sr-ms-map-af)#12.39.4.4/32 4 range 1
RP/0/0/CPU0:SP2R3(config-sr-ms-map-af)#router isis AS1239
RP/0/0/CPU0:SP2R3(config-isis)#address-family ipv4 unicast
RP/0/0/CPU0:SP2R3(config-isis-af)#segment-routing prefix-sid-map advertise-local
RP/0/0/CPU0:SP2R3(config-isis-af)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Now the traceroute from SP2R5 should show correct mapping provided to IGP domain by SR Mapping Server - traffic goes to domain edge labelled:

```
SP2R5-RR#traceroute 12.39.4.4 source lo0
Type escape sequence to abort.
Tracing the route to 12.39.4.4
VRF info: (vrf in name/id, vrf out name/id)
  1 12.39.35.3 [MPLS: Label 16004 Exp 0] 17 msec 15 msec 14 msec
  2 12.39.23.2 [MPLS: Label 16004 Exp 0] 13 msec 15 msec 11 msec
  3 12.39.12.1 [MPLS: Label 16004 Exp 0] 12 msec 11 msec 11 msec
  4 12.39.14.4 15 msec *  15 msec
```

Also, you can verify mapping being distributed in IS-IS within IGP domain on any router:

```
SP2R5-RR#sh isis segment-routing prefix-sid-map receive
Tag AS1239:

IS-IS Level-2 receive prefix-sid maps:
Host                    Prefix               SID Index    Range        Flags
SP2R3                   12.39.4.4/32         4            1
```

...and on IOS XR based device:

```
RP/0/0/CPU0:SP2R2#sh isis segment-routing prefix-sid-map active-policy

IS-IS AS1239 active policy
Prefix               SID Index    Range        Flags
12.39.4.4/32         4            1

Number of mapping entries: 1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5: New Segment Routing features

Please use following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch14.sr.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch14.sr.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch14-sr.unl`

Please notice, that version of IOS XR used in this chapter is 6.5.3. Version used in other labs (6.1.3) doesn't have features needed to complete those tasks.

Task 5.1: Simple SR TE with MPLS data plane with OSPF

In current releases of IOS XR, SR subsystem was modified to be aligned with additional capabilities SR TE provides. This means old "interface based" TE configuration was not enough anymore to handle all capabilities and is discontinued.



That's actually not true, at least for 6.x releases. If, and only if you boot up router with no Segment Routing configuration, you can still configure "old" way of doing TE tunnels and add "segment-routing" to path-option. As soon as router boots with segment-routing configured, you'll no longer be able to do that however. It's sort of one-time migration support.

New policy definiton syntax brings a lot of improvements.

Configuring SR TE policy under global SR TE configuration:

```
RP/0/0/CPU0:SP1R4(config)#segment-routing
RP/0/0/CPU0:SP1R4(config-sr)#traffic-eng
RP/0/0/CPU0:SP1R4(config-sr-te)#segment-list name SRTE45
RP/0/0/CPU0:SP1R4(config-sr-te-sl)#index 1 mpls label 16002
RP/0/0/CPU0:SP1R4(config-sr-te-sl)#index 2 mpls label 16003
RP/0/0/CPU0:SP1R4(config-sr-te-sl)#index 3 mpls label 16001
RP/0/0/CPU0:SP1R4(config-sr-te-sl)#index 4 mpls label 16005
RP/0/0/CPU0:SP1R4(config-sr-te-sl)#exit
RP/0/0/CPU0:SP1R4(config-sr-te)#exit
RP/0/0/CPU0:SP1R4(config-sr)#traffic-eng
RP/0/0/CPU0:SP1R4(config-sr-te)#policy PE-SR45
RP/0/0/CPU0:SP1R4(config-sr-te-policy)#color 1 end-point ipv4 33.56.5.5
RP/0/0/CPU0:SP1R4(config-sr-te-policy)#autoroute
RP/0/0/CPU0:SP1R4(config-sr-te-policy)#metric constant 1
RP/0/0/CPU0:SP1R4(config-sr-te-policy)#include all
RP/0/0/CPU0:SP1R4(config-sr-te-policy)#candidate-paths
RP/0/0/CPU0:SP1R4(config-sr-te-policy-path)#preference 10
RP/0/0/CPU0:SP1R4(config-sr-te-policy-path-pref)#explicit segment-list SRTE45
RP/0/0/CPU0:SP1R4(config-sr-te-policy-path-pref)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Resulting configuration should look like this:

```
segment-routing
 traffic-eng
  segment-list SRTE45
   index 1 mpls label 16002
   index 2 mpls label 16003
   index 3 mpls label 16001
   index 4 mpls label 16005
  !
  policy PE-SR45
   color 1 end-point ipv4 33.56.5.5
   autoroute
    metric constant 1
    include all
   !
   candidate-paths
    preference 10
     explicit segment-list SRTE45
     !
    !
   !
  !
 !
!
```

Configuration should result in SR TE subsystem reporting active tunnel. First, MPLS tunnels should be aware of the tunnel:

```
RP/0/0/CPU0:SP1R4#sh mpls forwarding tunnels
Tunnel                    Outgoing    Outgoing     Next Hop        Bytes
Name                      Label       Interface                    Switched
------------------------- ----------- ------------ --------------- ------------
sr-srte_c_1_ep_33.56.5.5(SR) 16003       Gi0/0/0/5    33.56.24.2      4711
```

Segment Routing TE system should also report TE policy as active for given endpoint:

```
RP/0/0/CPU0:SP1R4#sh segment-routing traffic-eng policy tabular

 Color             Endpoint  Admin   Oper              Binding
                             State  State                 SID
------ -------------------- ------ ------ --------------------
     1            33.56.5.5     up     up               104004
```

If you'd like to check Binding SID - it points directly to policy path itself:

```
RP/0/0/CPU0:SP1R4#sh mpls forwarding labels 104004
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
104004 Pop         No ID              srte_c_1_ep_ point2point     0
```

And at last, our SR TE policy should be visible in detail:

```
RP/0/0/CPU0:SP1R4#sh segment-routing traffic-eng policy

SR-TE policy database
---------------------

Color: 1, End-point: 33.56.5.5
  Name: srte_c_1_ep_33.56.5.5
  Status:
    Admin: up  Operational: up for 00:21:35 (since Jan  1 15:31:15.693)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Candidate-paths:
    Preference: 10 (configuration) (active)
      Name: PE-SR45
      Requested BSID: 45001
      Explicit: segment-list SRTE45 (valid)
        Weight: 1, Metric Type: TE
          16002
          16003
          16001
          16005
  Attributes:
    Binding SID: 45001
    Forward Class: 0
    Steering BGP disabled: no
    IPv6 caps enable: yes
```

Also, best path for 33.56.5.5 will be now reachable via SR TE path:

```
RP/0/0/CPU0:SP1R4#sh route ipv4 unicast 33.56.5.5

Routing entry for 33.56.5.5/32
  Known via "ospf AS3356", distance 110, metric 2, labeled SR, label redist non FIB,
type intra area
  Installed Jan  1 15:33:28.904 for 00:19:42
  Routing Descriptor Blocks
    33.56.5.5, from 33.56.5.5, via srte_c_1_ep_33.56.5.5
      Route metric is 2
  No advertising protos.
```

CEF should also agree where to direct traffic:

```
RP/0/0/CPU0:SP1R4#sh cef 33.56.5.5 detail
33.56.5.5/32, version 103, labeled SR, internal 0x1000001 0x83 (ptr 0xa151dabc) [1], 0x0
(0xa14ffdf0), 0xa20 (0xa185718c)
 Updated Jan  1 14:38:03.273
 Prefix Len 32, traffic index 0, precedence n/a, priority 1
 Extensions: context-label:16005
  gateway array (0xa145588c) reference count 1, flags 0x8028, source rib (7), 1 backups
                [2 type 1 flags 0x108401 (0xa1873c18) ext 0x0 (0x0)]
  LW-LDI[type=1, refc=1, ptr=0xa14ffdf0, sh-ldi=0xa1873c18]
  gateway array update type-time 1 Jan 17 14:38:03.273
 LDI Update time Jan 15 20:15:14.757
 LW-LDI-TS Jan 15 20:15:14.757
   via 33.56.5.5/32, srte_c_1_ep_33.56.5.5, 7 dependencies, weight 0, class 0 [flags
0x0]
    path-idx 0 NHID 0x0 [0xa13b45e0 0x0]
    next hop 33.56.5.5/32
    local adjacency
     labels imposed {None}

    Load distribution: 0 (refcount 2)

    Hash  OK  Interface                 Address
    0     Y   srte_c_1_ep_33.56.5.5     point2point
```

When we check traceroute to verify our explicit path, it should obviously match:

```
RP/0/0/CPU0:SP1R4#traceroute 33.56.5.5 source lo0

Type escape sequence to abort.
Tracing the route to 33.56.5.5

 1  33.56.24.2 [MPLS: Labels 16003/16001/16005 Exp 0] 9 msec  0 msec  9 msec
 2  33.56.23.3 [MPLS: Labels 16001/16005 Exp 0] 9 msec  9 msec  29 msec
 3  33.56.13.1 [MPLS: Label 16005 Exp 0] 9 msec  9 msec  9 msec
 4  33.56.15.5 9 msec  *  9 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

This way of configuring SR TE has additional features - which we will cover later in this chapter.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5.2: Simple SR TE with MPLS data plane with IS-IS

We can also check the same configuration concept while working with IS-IS. Once you configure traffic-engineering subsection on SP2R4, you should have following configuration:

SP2R4:

```
segment-routing
 traffic-eng
  segment-list SRTE45
   index 1 mpls label 16102
   index 2 mpls label 16103
   index 3 mpls label 16101
   index 4 mpls label 16105
  !
  policy PE-SR45
   color 1 end-point ipv4 12.39.5.5
   autoroute
    metric constant 1
    include all
   !
   candidate-paths
    preference 10
     explicit segment-list SRTE45
     !
    !
   !
  !
 !
!
```

And now for the same outputs as with OSPF:

```
RP/0/0/CPU0:SP2R4#sh mpls forwarding tunnels
Tunnel                     Outgoing    Outgoing     Next Hop        Bytes
Name                       Label       Interface                    Switched
-------------------------- ----------- ------------ --------------- ------------
sr-srte_c_1_ep_12.39.5(SR) 16103       Gi0/0/0/5    12.39.24.2      441
```

Tunnel table:

```
RP/0/0/CPU0:SP2R4#sh segment-routing traffic-eng policy tabular

 Color             Endpoint  Admin   Oper              Binding
                             State  State                 SID
------ -------------------- ------ ------ --------------------
     1            12.39.5.5     up     up               123353
```

SR policy in action:

```
RP/0/0/CPU0:SP2R4#sh segment-routing traffic-eng policy

SR-TE policy database
---------------------

Color: 1, End-point: 12.39.5.5
  Name: srte_c_1_ep_12.39.5.5
  Status:
    Admin: up  Operational: up for 00:02:44 (since Jan 15 20:22:51.305)
  Candidate-paths:
    Preference: 10 (configuration) (active)
      Name: PE-SR45
      Requested BSID: dynamic
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Explicit: segment-list SRTE45 (valid)
        Weight: 1, Metric Type: TE
          16102
          16103
          16101
          16105
  Attributes:
    Binding SID: 123353
    Forward Class: 0
    Steering BGP disabled: no
    IPv6 caps enable: yes
```

And let's check forwarding - routing entry:

```
RP/0/0/CPU0:SP2R4#sh route ipv4 unicast 12.39.5.5

Routing entry for 12.39.5.5/32
  Known via "isis AS1239", distance 115, metric 11, labeled SR, label redist non FIB,
type level-2
  Installed Jan 15 20:22:51.395 for 00:04:32
  Routing Descriptor Blocks
    12.39.5.5, from 12.39.5.5, via srte_c_1_ep_12.39.5.5
      Route metric is 11
  No advertising protos.
```

...and mandatory traceroute to check label stack for SR TE:

```
RP/0/0/CPU0:SP2R4#traceroute 12.39.5.5 source lo0

Type escape sequence to abort.
Tracing the route to 12.39.5.5

 1  12.39.24.2 [MPLS: Labels 16103/16101/16105 Exp 0] 9 msec  9 msec  9 msec
 2  12.39.23.3 [MPLS: Labels 16101/16105 Exp 0] 9 msec  0 msec  9 msec
 3  12.39.13.1 [MPLS: Label 16105 Exp 0] 9 msec  9 msec  9 msec
 4  12.39.15.5 9 msec  *  9 msec
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 6: Segment Routing additional features

Task 6.1: Traffic Collector

Segment Routing has built-in accounting capability. In short, it allows you to differentiate between external (outside of SR domain) and internal (all by default) interfaces, and then count traffic flowing to each Prefix SID. Data can be gathered and stored, or acquired by some external entity. Let's configure Traffic Collector feature within AS3356:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#traffic-collector
RP/0/0/CPU0:SP1R2(config-tc)#interface gi0/0/0/1
RP/0/0/CPU0:SP1R2(config-tc)#interface gi0/0/0/1.72
RP/0/0/CPU0:SP1R2(config-tc)#interface gi0/0/0/3
RP/0/0/CPU0:SP1R2(config-tc)#commit
```

Note that this example comes from SP1R2 - please adjust "external" interfaces definition on all other routers - R1, R3 and R4.

By default, you'll see collection interval of 1 minute and history kept for 48 hours:

```
RP/0/0/CPU0:SP1R1#show traffic-collector summary
Traffic Collector Summary:
  Statistics:
    Collection interval: 1 minutes (Running)
    History timeout    : 48 hours (Running)
    History size       : Last 5 collections
```

You can also check all interfaces defined as external:

```
RP/0/0/CPU0:SP1R1#sh traffic-collector external-interface
  Interface             Status
  --------------------  ----------------
  Gi0/0/0/1             Enabled
  Gi0/0/0/4             Enabled
```

Once AS 3356 will start to receive traffic crossing the internal/external border, you can start to monitor what's going on. For example, we can get quick look at what's going on in our domain (packets per second table):

```
RP/0/0/CPU0:SP1R1#sh traffic-collector ipv4 counters prefix
Prefix                          Label     Base rate     TM rate       State
                                          (Bytes/sec)   (Bytes/sec)
------------------------------  --------  ------------  ------------  ----------
33.56.2.2/32                    16002     4             2             Active
33.56.3.3/32                    16003     6             0             Active
33.56.4.4/32                    16004     16            0             Active
33.56.5.5/32                    16005     18            0             Active
```

If you want to ask about specific prefix, label or tunnel and show the whole history, it's easy as well - for example:

```
RP/0/0/CPU0:SP1R1#sh traffic-collector ipv4 counters label 16005 detail
Prefix: 33.56.5.5/32  SR Label: 16005 LDP Label: 1048577 State: Active
Base:
    Average over the last 5 collection intervals:
        Packet rate: 0 pps, Byte rate: 18 Bps
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    History of counters:
        19:32 - 19:33: Packets                    18, Bytes:                  1126
        19:31 - 19:32: Packets                    16, Bytes:                  1046
        19:30 - 19:31: Packets                    16, Bytes:                  1046
        19:29 - 19:30: Packets                    20, Bytes:                  1286
        19:28 - 19:29: Packets                    18, Bytes:                  1162

TM Counters:
    Average over the last 5 collection intervals:
        Packet rate: 0 pps, Byte rate: 0 Bps

    History of counters:
        19:32 - 19:33: Packets                     1, Bytes:                    64
        19:31 - 19:32: Packets                     1, Bytes:                    64
        19:30 - 19:31: Packets                     1, Bytes:                    64
        19:29 - 19:30: Packets                     1, Bytes:                    64
        19:28 - 19:29: Packets                     1, Bytes:                    64
```

That concludes Traffic Collector feature exercise.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 6.2: Segment Routing MPLS OAM

In releases later than IOS XR 6.1, MPLS OAMsubsystem got dedicated SR feature set. Once you activate MPLS OAM on router, you can use 'sr-mpls' option for typical OAM tools:

```
RP/0/0/CPU0:SP1R4#ping sr-mpls 33.56.5.5/32

Sending 5, 100-byte MPLS Echos to 33.56.5.5/32,
      timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
[...]

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/10 ms
```

And for MPLS OAM traceroute:

```
RP/0/0/CPU0:SP1R4#traceroute sr-mpls 33.56.5.5/32

Tracing MPLS Label Switched Path to 33.56.5.5/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
[...]

Type escape sequence to abort.

  0 33.56.34.4 MRU 1500 [Labels: 16005 Exp: 0]
L 1 33.56.34.3 MRU 1500 [Labels: implicit-null Exp: 0] 10 ms
! 2 33.56.35.5 10 ms
```

Both traceroute and ping can be asked to report specific IP responding to request by using 'verbose':

```
RP/0/0/CPU0:SP1R4#traceroute sr-mpls 33.56.5.5/32 verbose

Tracing MPLS Label Switched Path to 33.56.5.5/32, timeout is 2 seconds

[...]

Type escape sequence to abort.

  0 33.56.34.4 33.56.34.3 MRU 1500 [Labels: 16005 Exp: 0]
L 1 33.56.34.3 33.56.35.5 MRU 1500 [Labels: implicit-null Exp: 0] 0 ms, ret code 8
! 2 33.56.35.5 1 ms, ret code 3
```

You can extend traceroute to cover all possible paths - like this:

```
RP/0/0/CPU0:SP1R4#traceroute sr-mpls multipath 33.56.5.5/32 source 33.56.4.4

Starting LSP Path Discovery for 33.56.5.5/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
[...]

L!
Path 0 found,
 output interface GigabitEthernet0/0/0/0 nexthop 33.56.14.1
 source 33.56.4.4 destination 127.0.0.0
L!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Path 1 found,
 output interface GigabitEthernet0/0/0/2 nexthop 33.56.34.3
 source 33.56.4.4 destination 127.0.0.0

Paths (found/broken/unexplored) (2/0/0)
 Echo Request (sent/fail) (4/0)
 Echo Reply (received/timeout) (4/0)
 Total Time Elapsed 39 ms
```

And then, by using extra selector fields, you can force traceroute to look for ECMP multipaths between nodes on the way to destination. To discover up to two paths:

```
RP/0/0/CPU0:SP1R4#traceroute sr-mpls 33.56.5.5/32 destination 127.0.0.0 127.0.0.1

Tracing MPLS Label Switched Path to 33.56.5.5/32, timeout is 2 seconds

[...]

Destination address 127.0.0.0
  0 33.56.14.4 MRU 1500 [Labels: 16005 Exp: 0]
L 1 33.56.14.1 MRU 1500 [Labels: implicit-null Exp: 0] 10 ms
! 2 33.56.15.5 1 ms

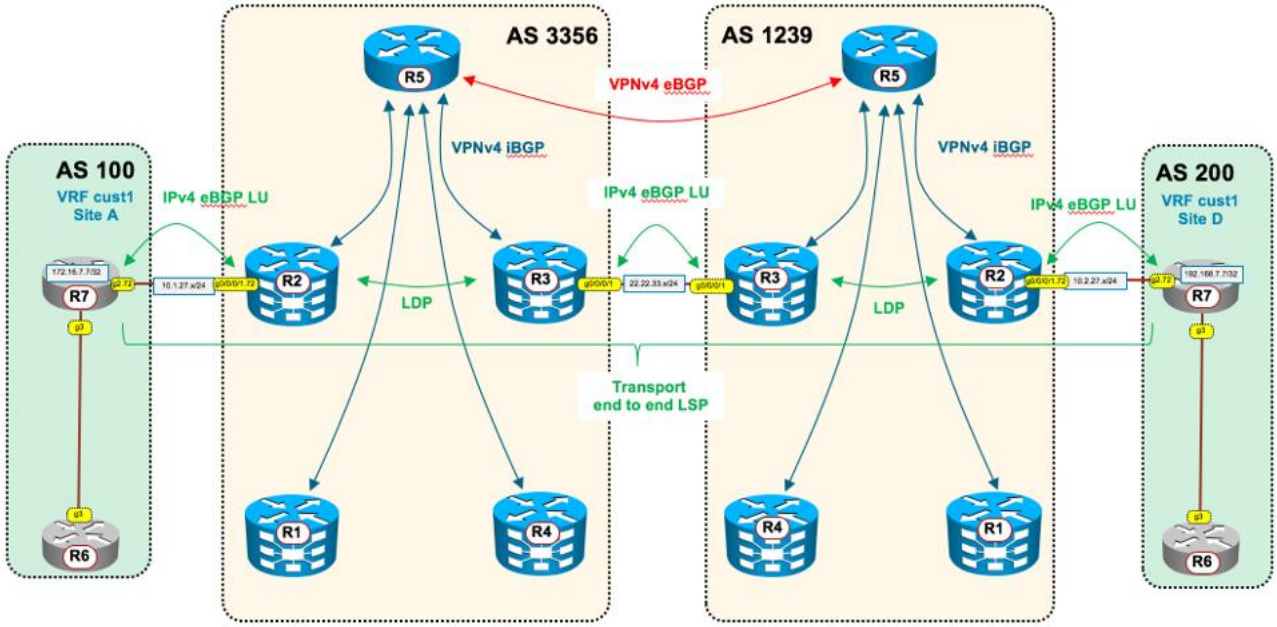
Destination address 127.0.0.1
  0 33.56.34.4 MRU 1500 [Labels: 16005 Exp: 0]
L 1 33.56.34.3 MRU 1500 [Labels: implicit-null Exp: 0] 0 ms
! 2 33.56.35.5 1 ms
```

If you have more potential paths, just increase last octet in the second destination specification (we're testing above up to two paths - .0 and .1).

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 15: IPv6 transition

Before starting this section make sure that there is a visibility of Loopback0 addresses of R7 routers between AS 100 and AS 200. Some tasks require the traffic in VRFs to be transported over AS 3356 and AS 1239. To do this a continuous LSP is needed as depicted in a figure attached below.



This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch15.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch15.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch15.unl`

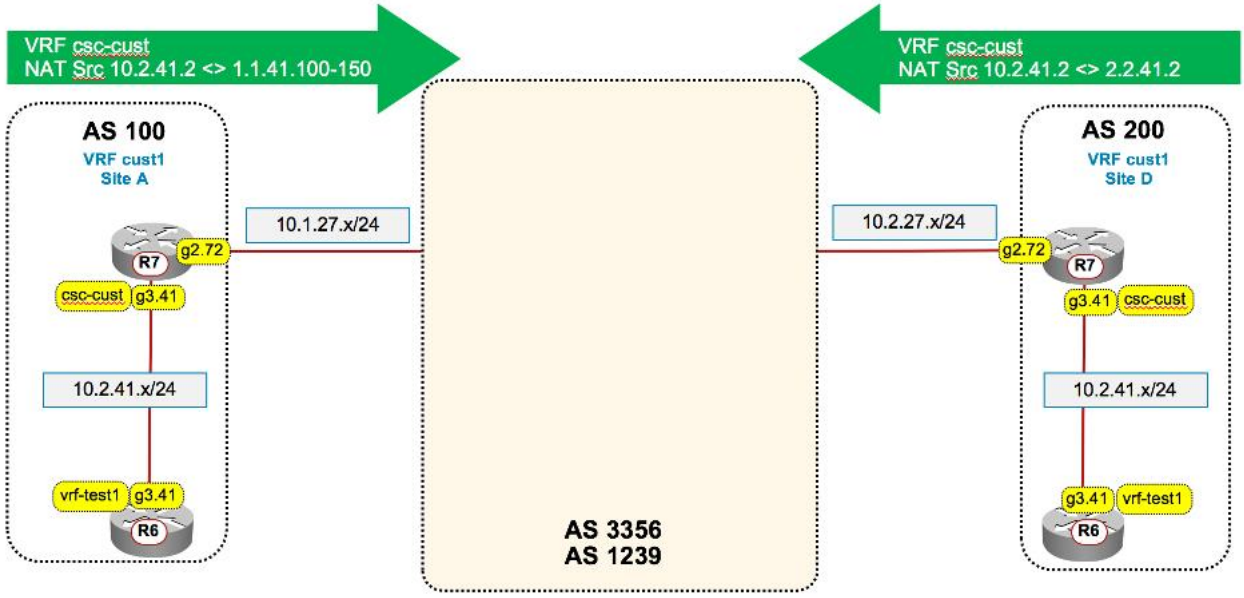
© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: Tunnelling and translation

You'll setup IPv6 transition methods by a usage of a tunnelling and translation. Some tunnels are part of the solution like in 6rd, DS-Lite or LISP. Some are added just to provide a direct connectivity between two sites. Used combinations in each of the task does not necessary mean you need to have them altogether. Like in the task 1.1 VRF-aware NAT44. You can have just NAT44 for a global routing table without a VRF.

Task 1.1: VRF-aware NAT44

There is the IPv4 address duplication between AS 100 and 200. The IP subnet 10.2.41.x/24 makes the communication impossible without readdressing or NAT. Configure a double NAT44 on R7 routers so R6 routers see each other with a unique IP address.



Add an additional separated subinterface to test the data path.

```
CEL6(config)#ip vrf vrf-test1
CEL6(config-vrf)#rd 41:1
CEL6(config)#interface GigabitEthernet3.41
CEL6(config-subif)# encapsulation dot1Q 41
CEL6(config-subif)# ip vrf forwarding vrf-test1
CEL6(config-subif)# ip address 10.2.41.2 255.255.255.0

CEL7(config)#interface GigabitEthernet3.41
CEL7(config-subif)# encapsulation dot1Q 41
CEL7(config-subif)# vrf forwarding csc-cust
CEL7(config-subif)# ip address 10.2.41.1 255.255.255.0

CER6(config)#ip vrf vrf-test1
CER6(config-vrf)#rd 41:1
CER6(config)#int g3.41
CER6(config-subif)#encapsulation dot1Q 41
CER6(config-subif)#ip vrf forwarding vrf-test1
CER6(config-subif)#ip add 10.2.41.2 255.255.255.0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Test a directly connected neighbor.

```
CER6#ping vrf vrf-test1 10.2.41.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.41.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 2/2/2 ms
```

Configure a default static route in the VRF vrf-test1 on R6 routers.

```
CEL6(config)#ip route vrf vrf-test1 0.0.0.0 0.0.0.0 10.2.41.1

CEL6#sh ip route vrf vrf-test1
[..]
Gateway of last resort is 10.2.41.1 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 10.2.41.1
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.2.41.0/24 is directly connected, GigabitEthernet3.41
L        10.2.41.2/32 is directly connected, GigabitEthernet3.41

CER6(config)#ip route vrf vrf-test1 0.0.0.0 0.0.0.0 10.2.41.1
```

Add and redistribute static entries toward the translated pools and accept those prefixes by the BGP filters.

```
CEL7(config)#ip route vrf csc-cust 1.1.41.0 255.255.255.0 null 0
CEL7(config)#ip prefix-list route72 seq 10 permit 1.1.41.0/24
CEL7(config)#router bgp 100
CEL7(config-router)#add ipv4 vrf csc-cust
CEL7(config-router-af)#redistribute static

CER7(config)#ip route vrf csc-cust 2.2.41.0 255.255.255.0 null 0
CER7(config)#ip prefix-list route72 seq 10 permit 2.2.41.0/24
CER7(config)#router bgp 200
CER7(config-router)#add ipv4 vrf csc-cust
CER7(config-router-af)#redistribute static
```

Verify a prefix advertisement.

```
CEL7#sh ip route vrf csc-cust
[..]
      1.0.0.0/24 is subnetted, 1 subnets
S        1.1.41.0 is directly connected, Null0
      2.0.0.0/24 is subnetted, 1 subnets
B        2.2.41.0 [20/0] via 192.168.7.72, 00:09:33
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.2.41.0/24 is directly connected, GigabitEthernet3.41
L        10.2.41.1/32 is directly connected, GigabitEthernet3.41
      17.0.0.0/32 is subnetted, 2 subnets
C        17.1.1.1 is directly connected, Loopback10
B        17.2.2.2 [20/0] via 192.168.7.72, 1w0d

CER7#sh ip bgp vpnv4 all
[..]
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf csc-cust)
 *>   1.1.41.0/24      172.16.7.72              0             0 100 ?
 *>   2.2.41.0/24      0.0.0.0                  0         32768 ?
 *    10.2.41.0/24     172.16.7.72              0             0 100 ?
 *>                    0.0.0.0                  0         32768 ?
 *>   17.1.1.1/32      172.16.7.72              0             0 100 ?
 *>   17.2.2.2/32      0.0.0.0                  0         32768 ?
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Add the NAT44 configuration. On CEL7 there is a NAT to a pool of IP addresses and on CER7 there is a 1:1 static NAT. Please note of the vrf keyword together with the nat statement which makes the NAT aware of the VRF traffic.

```
CEL7(config)#ip nat pool cgn 1.2.41.100 1.2.41.150 prefix-length 24
CEL7(config)#ip nat inside source list 102 pool cgn vrf csc-cust
CEL7(config)#ip access-list extended 102
CEL7(config-ext-nacl)#permit ip any any
```

Apply the NAT to interfaces.

```
CEL7(config)#int g3.41
CEL7(config-subif)#ip nat inside
CEL7(config-subif)#int g2.41
CEL7(config-subif)#ip nat outside
```

NAT settings on CER7.

```
CER7(config)#ip nat inside source static 10.2.41.2 2.2.41.2 vrf csc-cust
CER7(config)#int g3.41
CER7(config-subif)#ip nat inside
CER7(config)#int g2.72
CER7(config-subif)#ip nat out
```

Verification of IP reachability.

```
CEL6#ping vrf vrf-test1 2.2.41.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.41.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

It does not work. Let's see NAT statistics.

```
CEL7#sh ip nat stat
Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Outside interfaces:
  GigabitEthernet2.41
Inside interfaces:
  GigabitEthernet3.41
Hits: 0  Misses: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 102 pool cgn refcount 0
 pool cgn: id 1, netmask 255.255.255.0
        start 1.2.41.100 end 1.2.41.150
        type generic, total addresses 51, allocated 0 (0%), misses 0
nat-limit statistics:
 max entry: max allowed 0, used 0, missed 0
In-to-out drops: 0  Out-to-in drops: 0
Pool stats drop: 0  Mapping stats drop: 0
Port block alloc fail: 0
IP alias add fail: 0
Limit entry add fail: 0
```

There is a wrong outside interface. Instead of g2.72 there is g2.41. Let's fix it.

```
CEL7(config)#int g2.41
CEL7(config-subif)#no ip nat out
CEL7(config-subif)#int g2.72
CEL7(config-subif)#ip nat out
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Still no reachability.

```
CEL6#ping vrf vrf-test1 2.2.41.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.41.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

There is a dynamic NAT entry.

```
CEL7#sh ip nat trans
Pro  Inside global         Inside local          Outside local         Outside global
---  1.2.41.100            10.2.41.2             ---                   ---
icmp 1.2.41.100:22         10.2.41.2:22          2.2.41.2:22           2.2.41.2:22
Total number of translations: 2
```

But the translated IP address 1.2.41.100 is wrong. It should be 1.1.41.x. Let's see the configuration.

```
CEL7#sh run | i pool
ip nat pool cgn 1.2.41.100 1.2.41.150 prefix-length 24
ip nat inside source list 102 pool cgn vrf csc-cust
```

Correct this configuration.

```
CEL7(config)#no ip nat pool cgn 1.2.41.100 1.2.41.150 prefix-length 24
%Pool cgn in use, cannot destroy
CEL7(config)#no ip nat inside source list 102 pool cgn vrf csc-cust
%Dynamic mapping in use, cannot remove
```

To change the dynamic NAT configuration, you need to clear translations first of all.

```
CEL7#clear ip nat trans *

CEL7(config)#no ip nat pool cgn 1.2.41.100 1.2.41.150 prefix-length 24
CEL7(config)#no ip nat inside source list 102 pool cgn vrf csc-cust
CEL7(config)#ip nat pool cgn 1.1.41.100 1.1.41.150 prefix-length 24
CEL7(config)#ip nat inside source list 102 pool cgn vrf csc-cust
```

Verification of the IP reachability.

```
CEL6#ping vrf vrf-test1 2.2.41.2 sou 10.2.41.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.41.2, timeout is 2 seconds:
Packet sent with a source address of 10.2.41.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/26/32 ms

CEL6#
*May  7 18:12:43.284: ICMP: echo reply rcvd, src 2.2.41.2, dst 10.2.41.2, topology BASE,
dscp 0 topoid 3
*May  7 18:12:43.310: ICMP: echo reply rcvd, src 2.2.41.2, dst 10.2.41.2, topology BASE,
dscp 0 topoid 3
*May  7 18:12:43.335: ICMP: echo reply rcvd, src 2.2.41.2, dst 10.2.41.2, topology BASE,
dscp 0 topoid 3
*May  7 18:12:43.360: ICMP: echo reply rcvd, src 2.2.41.2, dst 10.2.41.2, topology BASE,
dscp 0 topoid 3
*May  7 18:12:43.393: ICMP: echo reply rcvd, src 2.2.41.2, dst 10.2.41.2, topology BASE,
dscp 0 topoid 3
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Capture packets to confirm a translation on a far end router.

```
CER6#sh monitor capture

Status Information for Capture cap
  Target Type:
 Interface: GigabitEthernet3.41, Direction: BOTH
   Status : Inactive
  Filter Details:
    Capture all packets
  Buffer Details:
   Buffer Type: LINEAR (default)
   Buffer Size (in MB): 10
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Maximum number of packets to capture per second: 1000
   Packet sampling rate: 0 (no sampling)

CER6#monitor capture cap start

CEL6#ping vrf vrf-test1 2.2.41.2 sou 10.2.41.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.41.2, timeout is 2 seconds:
Packet sent with a source address of 10.2.41.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/26/32 ms

CER6#monitor capture cap stop

CER6#sho monitor capture cap buffer brief
 ----------------------------------------------------------------------------
 #   size   timestamp     source              destination      dscp    protocol
 ----------------------------------------------------------------------------
[..]
  24   118 1031610.577017   1.1.41.100       ->  10.2.41.2        0  BE   ICMP
  25   118 1031610.577017   10.2.41.2        ->  1.1.41.100       0  BE   ICMP
  26   118 1031610.607014   1.1.41.100       ->  10.2.41.2        0  BE   ICMP
  27   118 1031610.608021   10.2.41.2        ->  1.1.41.100       0  BE   ICMP
  28   118 1031610.632022   1.1.41.100       ->  10.2.41.2        0  BE   ICMP
  29   118 1031610.632022   10.2.41.2        ->  1.1.41.100       0  BE   ICMP
  30   118 1031610.657015   1.1.41.100       ->  10.2.41.2        0  BE   ICMP
  31   118 1031610.657015   10.2.41.2        ->  1.1.41.100       0  BE   ICMP
  32   118 1031610.686020   1.1.41.100       ->  10.2.41.2        0  BE   ICMP
  33   118 1031610.686020   10.2.41.2        ->  1.1.41.100       0  BE   ICMP
```

The IP addresses are successfully translated.

```
CEL7#sh ip nat trans
Pro  Inside global         Inside local          Outside local         Outside global
---  1.1.41.100            10.2.41.2             ---                   ---
icmp 1.1.41.100:28         10.2.41.2:28          2.2.41.2:28           2.2.41.2:28
Total number of translations: 2

CER7#sh ip nat trans
Pro  Inside global         Inside local          Outside local         Outside global
---  2.2.41.2              10.2.41.2             ---                   ---
icmp 2.2.41.2:28           10.2.41.2:28          1.1.41.100:28         1.1.41.100:28
Total number of translations: 2
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

One you turn on of the CGN (Carrier Grade NAT) mode the status of the outside/translated address is not stored in the memory. This helps to scale up the NAT.

```
CEL7(config)#ip nat settings mode cgn

CEL7#sh ip nat trans
Pro  Inside global         Inside local          Outside local         Outside global
icmp 1.1.41.100:30         10.2.41.2:30          ---                   ---
---  1.1.41.100            10.2.41.2             ---                   ---
Total number of translations: 2
```

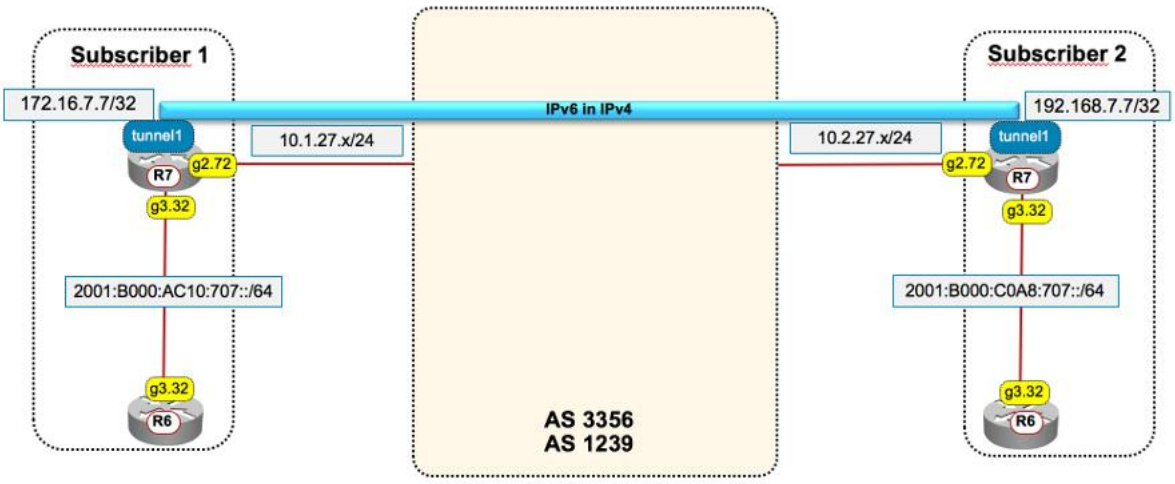


"You can configure CGN by using the ip nat settings mode cgn command. Use the ip nat settings mode default command to change to the default or traditional NAT operating mode. In the CGN mode, you cannot configure any NAT outside mappings. However, when you change from the default NAT mode to CGN mode, all existing outside mappings have to be removed. Use the no ip nat settings support mapping outside command to remove all outside mappings and to prevent any new outside mappings from being configured. You can also remove outside mappings by using the no form of commands used to configure NAT outside.
CGN increases the scalability of the number of NAT translations that can be supported because destination information is not stored."
source: http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_nat/configuration/xe-16/nat-xe-16-book/iadnat-cgn.html

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: 6rd

Configure 6rd tunnels which is a way to transport the IPv6 traffic over the IPv4 network. 6rd tunnels are not signalled with a remote end-point. This is a solution to add an IPv4 header with automatically calculated addresses based on the IPv6 local subnets.



Add 6rd tunnel interfaces and IPv6 addresses to a local interface.

CEL7:

```
!
interface Tunnel1
 no ip address
 no ip redirects
 ipv6 enable
 tunnel source Loopback0
 tunnel mode ipv6ip 6rd
 tunnel 6rd prefix 2001:B000::/32
!

CEL7(config)#ipv6 general-prefix gp1 6rd Tunnel1
CEL7(config)#interface GigabitEthernet3.32
CEL7(config-subif)#ipv6 address gp1 2001:B000::/64 eui-64
```

CER7:

```
!
interface Tunnel1
 no ip address
 no ip redirects
 ipv6 enable
 tunnel source Loopback0
 tunnel mode ipv6ip 6rd
 tunnel 6rd prefix 2001:B000::/32
!

CER7(config)#ipv6 general-prefix gp1 6rd Tunnel1
CER7(config)#interface GigabitEthernet3.32
CER7(config-subif)#ipv6 address gp1 2001:B000::/64 eui-64
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Add routing:

```
CEL7(config)#ipv6 route 2001:B000::/32 Tunnel1
CEL7(config)#ipv6 route 2001:B000::/32 Tunnel1
```

Verify IPv6 addresses. 2001:B000 is the prefix assigned to 6rd so a device is aware what to do with the traffic if a destination IP belongs to this prefix. The subscriber 1 has its site identified by the IPv4 address 172.16.7.7 which is the AC10:707 subnet portion of the address. The last 64 bits are the host part of the IPv6 address and it is generated by on a MAC address of a physical interface.

```
CEL7#sh ipv6 int g3.32
GigabitEthernet3.32 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::F816:3EFF:FEDB:8363
  No Virtual link-local address(es):
  General-prefix in use for addressing
  Global unicast address(es):
    2001:B000:AC10:707:F816:3EFF:FEDB:8363, subnet is 2001:B000:AC10:707::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FFDB:8363
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium

CER7#sh ipv6 int g3.32
GigabitEthernet3.32 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::F816:3EFF:FE18:8806
  No Virtual link-local address(es):
  General-prefix in use for addressing
  Global unicast address(es):
    2001:B000:C0A8:707:F816:3EFF:FE18:8806, subnet is 2001:B000:C0A8:707::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF18:8806
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.
```

Verify the 6rd prefix and the subnet portion which identifies a site.

```
CEL7#sh tunnel 6rd tunnel 1
Interface Tunnel1:
  Tunnel Source: 172.16.7.7
  6RD: Operational, V6 Prefix: 2001:B000::/32
       V4 Prefix, Length: 0, Value: 0.0.0.0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
       V4 Suffix, Length: 0, Value: 0.0.0.0
  General Prefix: 2001:B000:AC10:707::/64

CER7#sh tunnel 6rd tunnel 1
Interface Tunnel1:
  Tunnel Source: 192.168.7.7
  6RD: Operational, V6 Prefix: 2001:B000::/32
       V4 Prefix, Length: 0, Value: 0.0.0.0
       V4 Suffix, Length: 0, Value: 0.0.0.0
  General Prefix: 2001:B000:C0A8:707::/64
```

Verification that the remote site is identified correctly.

```
CEL7#sh tunnel 6rd destination 2001:B000:C0A8:707:F816:3EFF:FE18:8806 t1
Interface: Tunnel1
6RD Prefix: 2001:B000:C0A8:707:F816:3EFF:FE18:8806
Destination: 192.168.7.7

CEL7#sh tunnel 6rd prefix 192.168.7.7 t1
Interface: Tunnel1
Destination: 192.168.7.7
6RD Prefix: 2001:B000:C0A8:707::
```

Verification of the IP reachability.

```
CEL7#ping 2001:B000:C0A8:707:F816:3EFF:FE18:8806
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:B000:C0A8:707:F816:3EFF:FE18:8806, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 22/33/77 ms
```

A packet capture.

```
No.   Time      Source              Destination         Protocol Length Info
  229 10.2550… 20.40.27.7          224.0.0.10          EIG…    317 Update
  230 10.2669… 20.40.27.2          20.40.27.7          EIG…     54 Hello (Ack)
  231 10.3103… 2001:b000:ac…       2001:b000:c0…       ICM…    134 Echo (ping) request id=0x14ce, seq=0, hop limit=64
  232 10.3113… 2001:b000:c0…       2001:b000:ac…       ICM…    134 Echo (ping) reply id=0x14ce, seq=0, hop limit=64 (
  233 10.3331… 2001:b000:ac…       2001:b000:c0…       ICM…    134 Echo (ping) request id=0x14ce, seq=1, hop limit=64
  234 10.3341… 2001:b000:c0…       2001:b000:ac…       ICM…    134 Echo (ping) reply id=0x14ce, seq=1, hop limit=64 (
  235 10.3556… 2001:b000:ac…       2001:b000:c0…       ICM…    134 Echo (ping) request id=0x14ce, seq=2, hop limit=64
▸ Frame 231: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits)
▸ Ethernet II, Src: fa:16:3e:d3:f9:4e (fa:16:3e:d3:f9:4e), Dst: fa:16:3e:e0:94:b8 (fa:16:3e:e0:94:b8)
▸ Internet Protocol Version 4, Src: 172.16.7.7, Dst: 192.168.7.7
▾ Internet Protocol Version 6, Src: 2001:b000:ac10:707:f816:3eff:fedb:8363, Dst: 2001:b000:c0a8:707:f816:
    0110 .... = Version: 6
  ▸ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 60
    Next header: ICMPv6 (58)
    Hop limit: 64
    Source: 2001:b000:ac10:707:f816:3eff:fedb:8363
    Destination: 2001:b000:c0a8:707:f816:3eff:fe18:8806
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
```

You can change the way how the subnet portion of the IPv6 address is calculated by declaring common bits between sites. The addresses 172.16.7.7 and 192.168.7.7 which identify locations have just one bit in common. The decimal value of this bit is 128. This makes a change in the IPv6 addresses on 6rd interfaces.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CEL7(config)#int t1
CEL7(config-if)#tunnel 6rd ipv4 prefix-len 1

CEL7#sh tunnel 6rd destination 2001:B000:C0A8:707:F816:3EFF:FE18:8806 t1
Interface: Tunnel1
6RD Prefix: 2001:B000:C0A8:707:F816:3EFF:FE18:8806
Destination: 224.84.3.131
```

As you can see a general IPv6 prefix is changed from 2001:B000:C0A8:707::/64 to 2001:B000:5820:E0E::/63.

```
CEL7#sh tunnel 6rd t1
Interface Tunnel1:
  Tunnel Source: 172.16.7.7
  6RD: Operational, V6 Prefix: 2001:B000::/32
       V4 Prefix, Length: 1, Value: 128.0.0.0
       V4 Suffix, Length: 0, Value: 0.0.0.0
  General Prefix: 2001:B000:5820:E0E::/63

CEL7#sh ipv6 int g3.32
GigabitEthernet3.32 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::F816:3EFF:FEDB:8363
  No Virtual link-local address(es):
  General-prefix in use for addressing
  Global unicast address(es):
    2001:B000:5820:E0E:F816:3EFF:FEDB:8363, subnet is 2001:B000:5820:E0E::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FFDB:8363
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.
```

To make 6rd working the common bits must be configured on all sites.

```
CER7(config)#int t1
CER7(config-if)#tunnel 6rd ipv4 prefix-len 1

CER7#sh ipv6 int g3.32
GigabitEthernet3.32 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::F816:3EFF:FE18:8806
  No Virtual link-local address(es):
  General-prefix in use for addressing
  Global unicast address(es):
    2001:B000:8150:E0E:F816:3EFF:FE18:8806, subnet is 2001:B000:8150:E0E::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF18:8806
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
```

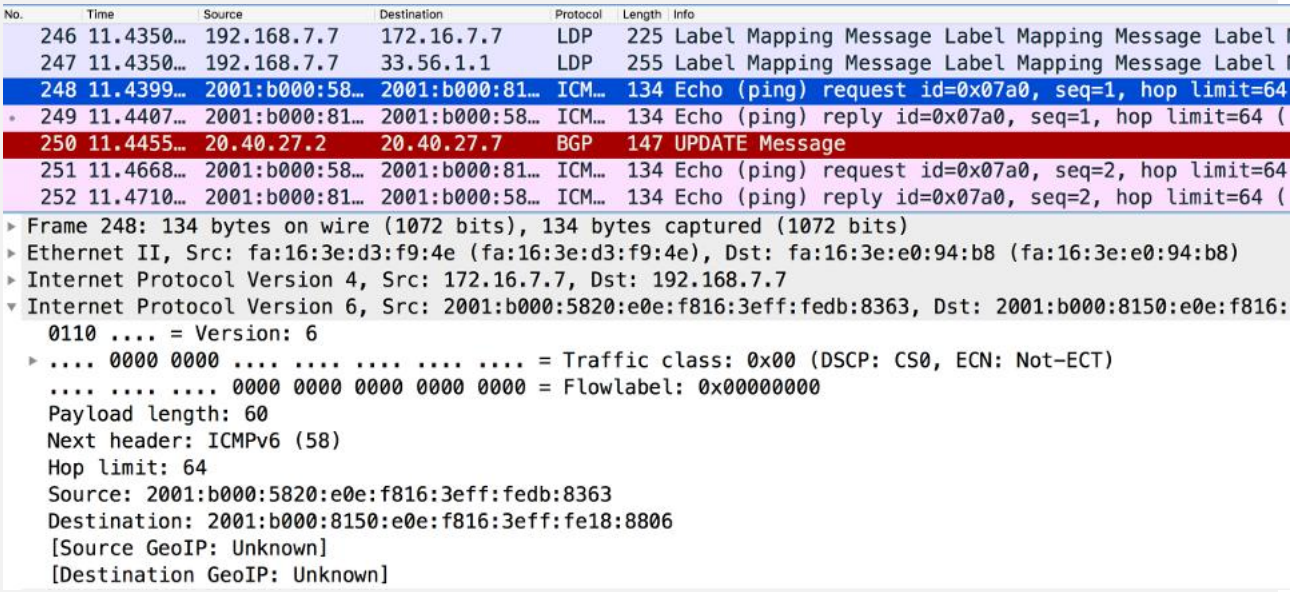

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.
```

To check the current location of the IPv6 address you need to use a new end-point address.

```
CEL7#sh tunnel 6rd destination 2001:B000:8150:E0E:F816:3EFF:FE18:8806 t1
Interface: Tunnel1
6RD Prefix: 2001:B000:8150:E0E:F816:3EFF:FE18:8806
Destination: 192.168.7.7

CEL7#ping 2001:B000:8150:E0E:F816:3EFF:FE18:8806
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:B000:8150:E0E:F816:3EFF:FE18:8806, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 23/25/35 ms
```

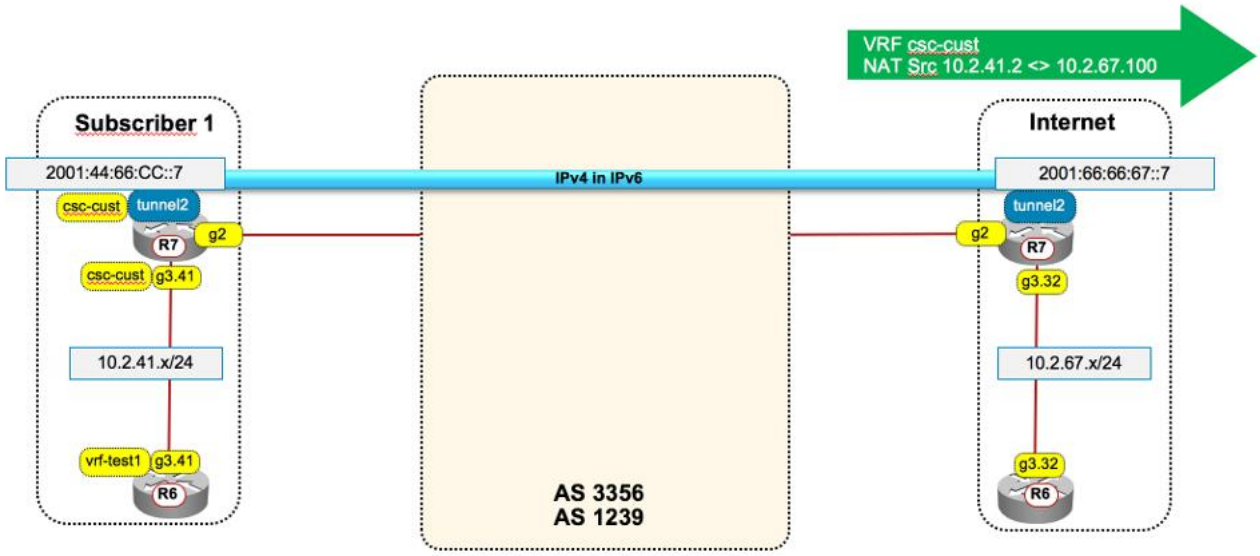


As you can see the IPv6 traffic is encapsulated into the IPv4 header. The IPv4 addresses are the 6rd tunnel end-points.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3: Pseudo DS-Lite

DS-Lite is not yet supported on IOS XE. It is supported on IOS XR but it requires a service module. Instead of this configure a simulated DS-Lite which is represented by an IPv6 in IPv4 tunnel.



This task requires the IPv6 reachability of the following addresses between AS 100 and AS 200. Use your knowledge to provide this connectivity through the L3VPN network.

The expected result.

```
CEL7#sh ipv6 route 2001:66:66:67::7
Routing entry for 2001:66:66:67::/64
  Known via "bgp 100", distance 20, metric 0, type external
  Route count is 1/1, share count 0
  Routing paths:
    FE80::2, GigabitEthernet2
      MPLS label: nolabel
      From 2001:40:60:27::2
      Last updated 2d10h ago

CER7#sh ipv6 route 2001:44:66:CC::7
Routing entry for 2001:44:66:CC::7/128
  Known via "bgp 200", distance 20, metric 0, type external
  Route count is 1/1, share count 0
  Routing paths:
    FE80::2, GigabitEthernet2
      MPLS label: nolabel
      From 2001:20:40:27::2
      Last updated 2d10h ago

CEL7#ping 2001:66:66:67::7 sou 2001:44:66:CC::7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:66:66:67::7, timeout is 2 seconds:
Packet sent with a source address of 2001:44:66:CC::7
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 66/96/140 ms
```

Configure DS-Lite tunnels.

```
CEL7(config)#interface tunnel2
CEL7(config-if)# ip address 22.1.1.1 255.255.255.0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CEL7(config-if)# tunnel source 2001:44:66:CC::7
CEL7(config-if)# tunnel mode ipv6
CEL7(config-if)# tunnel destination 2001:66:66:67::7

CER7(config)#interface tunnel2
CER7(config-if)# ip address 22.1.1.2 255.255.255.0
CER7(config-if)# tunnel source 2001:66:66:67::7
CER7(config-if)# tunnel mode ipv6
CER7(config-if)# tunnel destination 2001:44:66:CC::7

CEL7#sh int t2
Tunnel2 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 22.1.1.1/24
  MTU 1460 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel linestate evaluation up
  Tunnel source 2001:44:66:CC::7, destination 2001:66:66:67::7
  Tunnel protocol/transport IPv6
  Tunnel TTL 255
  Path MTU Discovery, ager 10 mins, min MTU 1280
  Tunnel transport MTU 1460 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input never, output 01:17:13, output hang never
  Last clearing of "show interface" counters 01:24:28
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     245 packets input, 29988 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     282 packets output, 34868 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out

CER7#sh int t2
Tunnel2 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 22.1.1.2/24
  MTU 1460 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel linestate evaluation up
  Tunnel source 2001:66:66:67::7, destination 2001:44:66:CC::7
  Tunnel protocol/transport IPv6
  Tunnel TTL 255
  Path MTU Discovery, ager 10 mins, min MTU 1280
  Tunnel transport MTU 1460 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input never, output 01:22:26, output hang never
  Last clearing of "show interface" counters 01:25:37
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     296 packets input, 36384 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
     294 packets output, 35932 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
```

Add a VRF to the tunnel. On CEL7 the tunnel end-point is within a VRF. The other end of this tunnel on CER7 is in the global routing table (GRT). This is not required for DS-Lite but this is one of options to provide a leaking between VRF and GRT.

```
CEL7(config)#int t2
CEL7(config-if)# vrf forwarding csc-cust
CEL7(config-if)# ip address 22.1.1.1 255.255.255.0
```

Configure a dynamic way to exchange prefix information.

```
CEL7(config)#int t2
CEL7(config-if)# ip ospf 2 area 0
CEL7(config)#int g3.41
CEL7(config-subif)#ip ospf 2 ar 0

CER7(config)#int t2
CER7(config-if)# ip ospf 2 area 0
CER7(config)#int g3.32
CER7(config-subif)#ip ospf 2 are 0
```

Verify OSPF and received routes.

```
CEL7#sh ip ospf 2 nei

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.7.73      0   FULL/  -        00:00:38    22.1.1.2        Tunnel2

CER7#sh ip ospf 2 nei

Neighbor ID     Pri   State           Dead Time   Address         Interface
17.1.1.1          0   FULL/  -        00:00:31    22.1.1.1        Tunnel2

CEL7#sh ip route vrf csc-cust ospf
[..]
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O        10.2.67.0/24 [110/1001] via 22.1.1.2, 00:13:59, Tunnel2

CER7#sh ip route os
[..]
      10.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
O        10.2.41.0/24 [110/1001] via 22.1.1.1, 00:13:55, Tunnel2
      33.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O        33.56.1.1/32 [110/2] via 33.1.1.1, 2d10h, BDI33
```

Add NAT44 to simulate the exit to the extranet or the Internet.

```
CER7(config)#int t2
CER7(config-if)# ip nat inside
CER7(config-if)#int g3.32
CER7(config-subif)#ip nat outside
CER7(config-subif)#ip nat inside source static 10.2.41.2 10.2.67.100
```

Verify the IP reachability.

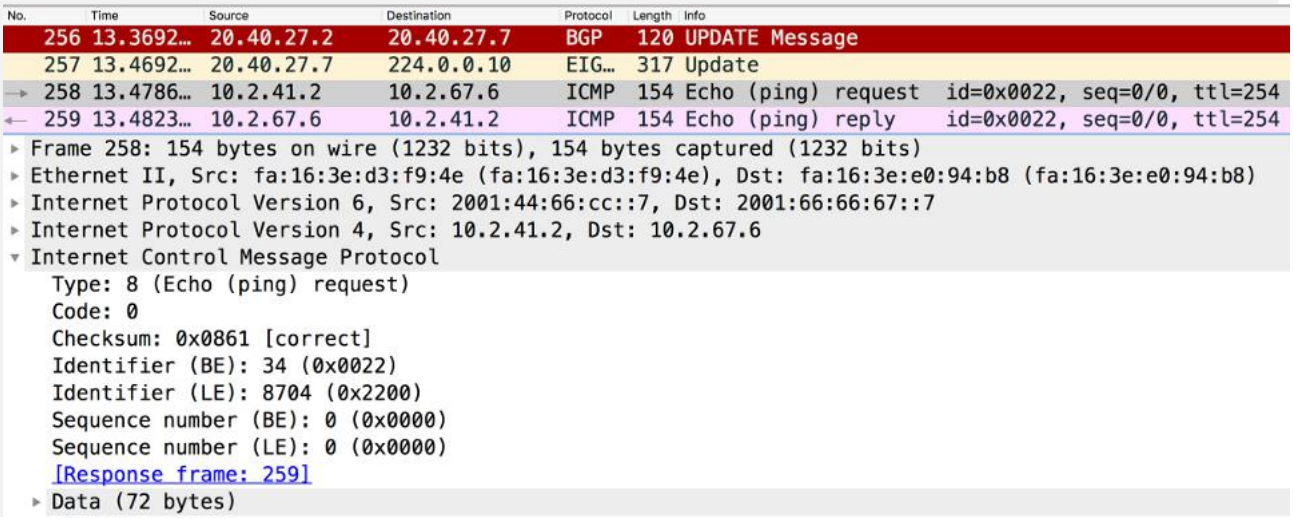
```
CEL6#ping vrf vrf-test1 10.2.67.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.67.6, timeout is 2 seconds:
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 72/98/114 ms
```

See NAT translations.

```
CER7#sh ip nat trans
Pro  Inside global         Inside local          Outside local         Outside global
---  10.2.67.100           10.2.41.2             ---                   ---
---  2.2.41.2              10.2.41.2             ---                   ---
icmp 10.2.67.100:32        10.2.41.2:32          10.2.67.6:32          10.2.67.6:32
Total number of translations: 3
```

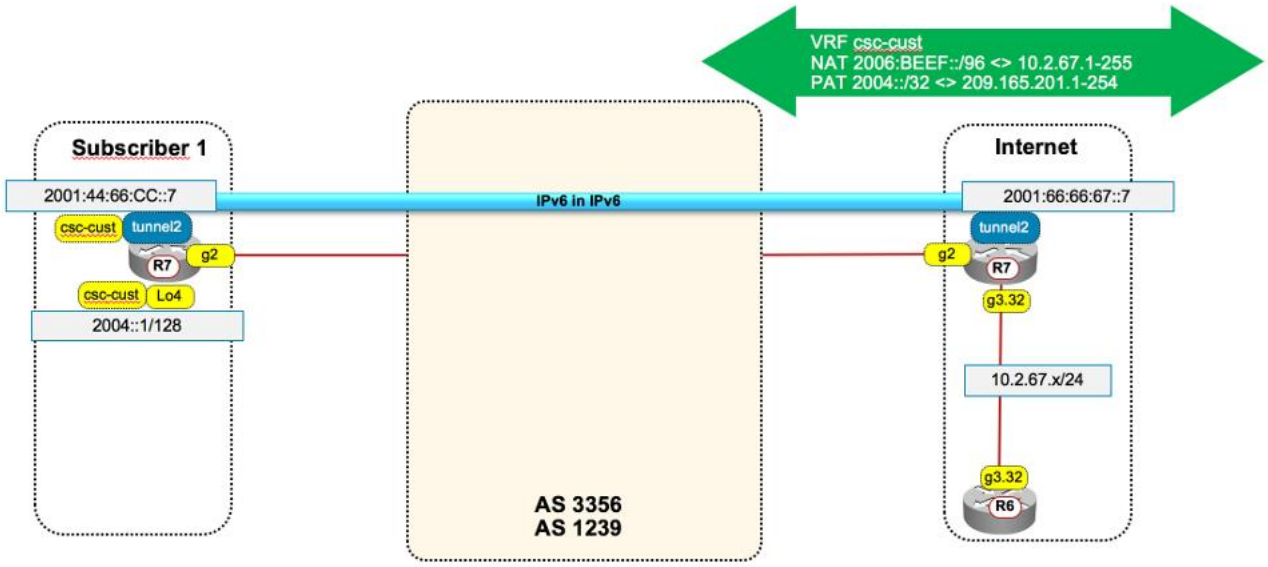


This packet capture shows not yet translated packet no. 258 which is crossing the IPv6 network by tunnelling IPv4 in IPv6.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4: Stateful NAT64

Configure a stateful NAT64 which translates the IPv6 addresses to IPv4.



Add the IPv6 address family to the existing VRF definition and assign the VRF to Lo4. Enable OSPF on Lo4 and tunnel2 interfaces on R7 routers.

```
CEL7(config)#vrf definition csc-cust
CEL7(config-vrf)# address-family ipv6

CEL7(config)#interface Loopback4
CEL7(config-if)# vrf forwarding csc-cust
CEL7(config-if)# ipv6 address 2004::1/128
CEL7(config-if)# ospfv3 2 ipv6 area 0

CEL7(config)#interface t2
CEL7(config-if)# ipv6 enable
CEL7(config-if)# ospfv3 2 ipv6 area 0

CER7(config)#interface t2
CER7(config-if)# ipv6 enable
CER7(config-if)# ospfv3 2 ipv6 area 0
```

Verify OSPF route advertisement.

```
CEL7#sh ospfv3 vrf csc-cust nei

          OSPFv3 2 address-family ipv6 vrf csc-cust (router-id 17.1.1.1)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
192.168.7.73      0   FULL/  -        00:00:39    30              Tunnel2

CER7#sh ospfv3 nei

          OSPFv3 2 address-family ipv6 (router-id 192.168.7.73)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
17.1.1.1          0   FULL/  -        00:00:33    31              Tunnel2
```

Configure NAT64. The public IPv4 end-points are translated to addresses derived from the prefix 2006:BEEF::/96. The SP based IPv6 subscribers are

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

translated to the IPv4 pool 209.165.201.1-254. This pool can be overloaded so the N:1 translation based on a port address translation (PAT).

```
CER7(config)#ipv6 access-list nat64-acl
CER7(config-ipv6-acl)# permit ipv6 2004::/32 any

CER7(config)#nat64 prefix stateful 2006:BEEF::/96
CER7(config)#nat64 v4 pool pool1 209.165.201.1 209.165.201.254
CER7(config)#nat64 v6v4 list nat64-acl pool pool1 overload

CER7(config)#interface t2
CER7(config-if)# nat64 enable
CER7(config)#interface g3.32
CER7(config-subif)# nat64 enable
```

Advertise a translated IPv4 prefix to a subscriber.

```
CER7(config)#router ospfv3 2
CER7(config-router)#address-family ipv6 unicast
CER7(config-router-af)#redistribute static
```

Set a static route pointing at the translated IPv4 prefix.

```
CER6(config)#ip route 209.165.201.0 255.255.255.0 10.2.67.7
```

Verification of the route visibility.

```
CEL7#sh ipv6 route vrf csc-cust
IPv6 Routing Table - csc-cust - 5 entries
[..]
OE2 64:FF9B::/96 [110/20]
     via FE80::21E:49FF:FE1C:A200, Tunnel2
OE2 2001:B000::/32 [110/20]
     via FE80::21E:49FF:FE1C:A200, Tunnel2
LC  2004::1/128 [0/0]
     via Loopback4, receive
OE2 2006:BEEF::/96 [110/20]
     via FE80::21E:49FF:FE1C:A200, Tunnel2
L   FF00::/8 [0/0]
     via Null0, receive

CEL7#sh ipv6 route vrf csc-cust 2006:BEEF::a02:4306
Routing entry for 2006:BEEF::/96
  Known via "ospf 2", distance 110, metric 20, type extern 2
  Route count is 1/1, share count 0
  Routing paths:
    FE80::21E:49FF:FE1C:A200, Tunnel2
      From FE80::21E:49FF:FE1C:A200
      Last updated 05:32:42 ago

CER7#sh ipv6 route 2004::1
Routing entry for 2004::1/128
  Known via "ospf 2", distance 110, metric 1000, type intra area
  Route count is 1/1, share count 0
  Routing paths:
    FE80::21E:49FF:FEDC:A400, Tunnel2
      From FE80::21E:49FF:FEDC:A400
      Last updated 05:35:22 ago
```

Check IP reachability.

```
CER6#monitor cap cap2 int g3.32 both
CER6#monitor cap cap2 match ipv4 209.165.201.0/24 any
A filter is already attached to the capture . Replace with specified filter?[confirm]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CER6#monitor cap cap2 start
Started capture point : cap2

CEL7#ping vrf csc-cust ipv6 2006:BEEF::A02:4306
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2006:BEEF::A02:4306, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 87/109/133 ms
```

No.	Time	Source	Destination	Protocol	Length	Info
246	12.1052…	2006:beef::a…	2004::1	ICM…	154	Echo (ping) reply id=0x2368, seq=1, hop limit=254
247	12.1912…	2004::1	2006:beef::a…	ICM…	154	Echo (ping) request id=0x2368, seq=2, hop limit=64
248	12.1917…	172.16.7.7	192.168.7.72	LDP	80	Hello Message
249	12.2152…	2006:beef::a…	2004::1	ICM…	154	Echo (ping) reply id=0x2368, seq=2, hop limit=254
250	12.2833…	2004::1	2006:beef::a…	ICM…	154	Echo (ping) request id=0x2368, seq=3, hop limit=64
251	12.2984…	2006:beef::a…	2004::1	ICM…	154	Echo (ping) reply id=0x2368, seq=3, hop limit=254

```
▸ Frame 247: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits)
▸ Ethernet II, Src: fa:16:3e:d3:f9:4e (fa:16:3e:d3:f9:4e), Dst: fa:16:3e:e0:94:b8 (fa:16:3e:e0:94:b8)
▸ Internet Protocol Version 6, Src: 2001:44:66:cc::7, Dst: 2001:66:66:67::7
▸ Internet Protocol Version 6, Src: 2004::1, Dst: 2006:beef::a02:4306
▸ Internet Control Message Protocol v6
```

This capture shows that the IPv6 ICMP packet is encapsulated into the IPv6 packet which is the tunnel2.

```
CER6#monitor cap cap2 stop
Stopped capture point : cap2
```

To see translated addresses display a packet capture buffer.

```
CER6#show monitor cap cap2 buffer brief
 ------------------------------------------------------------------------------
 #   size   timestamp     source               destination      dscp    protocol
 ------------------------------------------------------------------------------
   0   98     0.000000    209.165.201.1     ->  10.2.67.6        0  BE   ICMP
   1   98     0.127984    209.165.201.1     ->  10.2.67.6        0  BE   ICMP
   2   98     0.246981    209.165.201.1     ->  10.2.67.6        0  BE   ICMP
   3   98     0.326979    209.165.201.1     ->  10.2.67.6        0  BE   ICMP
   4   98     0.431969    209.165.201.1     ->  10.2.67.6        0  BE   ICMP
```

The status of NAT is kept in a table.

```
CER7#sh nat64 trans

Proto  Original IPv4          Translated IPv4
       Translated IPv6        Original IPv6
------------------------------------------------------------------------------

icmp   10.2.67.6:1            [2006:beef::a02:4306]:8611
       209.165.201.1:1        [2004::1]:8611

Total number of translations: 1
```

One more ping creates one more NAT state.

```
CEL7#ping vrf csc-cust ipv6 2006:BEEF::A02:4307
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2006:BEEF::A02:4307, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/95/190 ms

CER7#sh nat64 trans

Proto  Original IPv4          Translated IPv4
       Translated IPv6        Original IPv6
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
--------------------------------------------------------------------------------

icmp   10.2.67.6:1           [2006:beef::a02:4306]:8611
       209.165.201.1:1       [2004::1]:8611
icmp   10.2.67.7:2           [2006:beef::a02:4307]:308
       209.165.201.1:2       [2004::1]:308

Total number of translations: 2
```

A more detailed output contains also a session date.

```
CER7#sh nat64 trans verbose

Proto  Original IPv4         Translated IPv4
       Translated IPv6       Original IPv6
--------------------------------------------------------------------------------

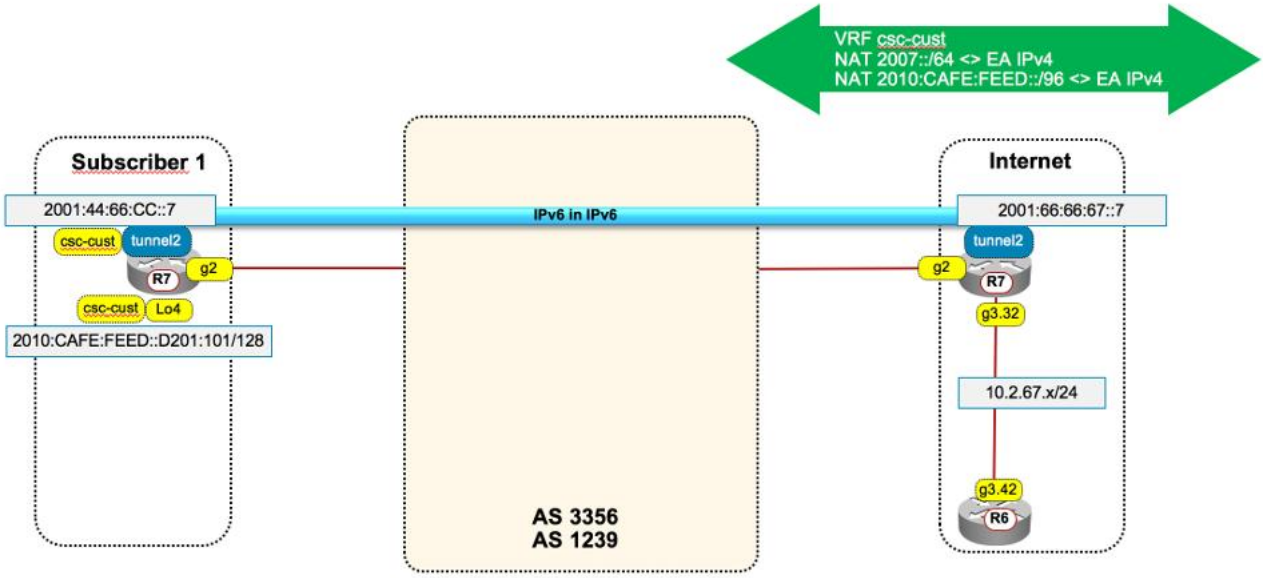
icmp   10.2.67.6:1           [2006:beef::a02:4306]:953
       209.165.201.1:1       [2004::1]:953
         created: 01 jan 2017 00:31:18, last-used:   01 jan 2017 00:31:08,
           inactivity-time:     00:00:56
         flags: none
         entry-id: 0xeb79e9e0, use-count: 1
icmp   10.2.67.7:2           [2006:beef::a02:4307]:5352
       209.165.201.1:2       [2004::1]:5352
         created: 01 jan 2017 00:31:19, last-used:   01 jan 2017 00:31:20,
           inactivity-time:     00:00:58
         flags: none
         entry-id: 0xeb79ea80, use-count: 1

Total number of translations: 2
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.5: Stateless NAT64

Configure the stateless version of NAT64.



NAT64 configuration.

```
CER7(config)#nat64 prefix stateless v4v6 2007::/64
CER7(config)#int t2
CER7(config-if)#nat64 prefix stateless v6v4 2010:CAFE:FEED::/96
```

There are two stateless IPv4-translatable IPv6 prefixes. This means that based on the IPv6 address a router can calculate algorithmically the IPv4 address and vice versa. The 2007::/64 prefix is used by the external IPv4 nodes. The 2010:CAFE:FEED::/96 prefix is the internal prefix which belongs to a Service Provider and this is translated to the IPv4 which is embedded into the IPv6 address. The algorithm is described by RFC 6052.



From RFC 6052 - IPv4-converted IPv6 addresses and IPv4-translatable IPv6 addresses follow the same format, described here as the IPv4-embedded IPv6 address Format. IPv4-embedded IPv6 addresses are composed of a variable-length prefix, the embedded IPv4 address, and a variable-length suffix, as presented in the following diagram, in which PL designates the prefix length:

```
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|PL| 0-------------32--40--48--56--64--72--80--88--96--104---------|
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|32|     prefix    |v4(32)         | u | suffix                    |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|40|     prefix        |v4(24)     | u |(8)| suffix                |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|48|     prefix            |v4(16) | u | (16)  | suffix            |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|56|     prefix                |(8)| u |  v4(24)   | suffix        |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|64|     prefix                    | u |   v4(32)      | suffix    |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|96|     prefix                                    |    v4(32)     |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 1

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

In these addresses, the prefix shall be either the "Well-Known Prefix" or a "Network-Specific Prefix" unique to the organization deploying the address translators. The prefixes can only have one of the following lengths: 32, 40, 48, 56, 64, or 96. (The Well-Known Prefix is 96 bits long, and can only be used in the last form of the table.) The IPv4 address is encoded following the prefix, most significant bits first. Depending of the prefix length, the 4 octets of the address may be separated by the reserved octet "u", whose 8 bits MUST be set to zero. When the prefix is 64 bits long, the IPv4 address is encoded in positions 72 to 103.

Configure routing. The first statement is a special route entry for NAT64.

```
CER7(config)#nat64 route 210.1.1.0/24 Tunnel2

CER7#sh nat64 routes

IPv4 Prefix        Adj. Address    Enabled Output IF
   Global IPv6 Prefix

210.1.1.0/24       0.0.0.1         TRUE    Tu2
```

You can notice the NVI type of interface used to direct the traffic for translation.

```
CER7#sh ip route 210.1.1.0
Routing entry for 210.1.1.0/24
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 0.0.0.1, via NVI0
      Route metric is 0, traffic share count is 1
```



"NAT Virtual Interface (NVI) feature allows NAT traffic flows on the virtual interface, eliminating the need to specify inside and outside domains. When a domain is specified, translation rules are applied either before or after the route decisions, depending on the traffic flow from inside to outside or outside to inside. The translation rules are applied only after the route decision for an NVI.

NAT Virtual Interface has the following benefits:
- A NAT table is maintained per interface for better performance and scalability.

Domain-specific NAT configurations can be eliminated."

source: http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_nat/configuration/12-4t/nat-12-4t-book/iadnat-addr-consv.html

```
CER6(config)#ip route 210.1.1.0 255.255.255.0 10.2.67.7
```

As the public IPv4 address is set to 210.1.1.x you can algorithmically calculate what will be the IPv6 address based on the IPv4. For 210.1.1.1 the IPv6 Embedded Address (EA) is D201:101. This is merged with the stateless prefix 2010:CAFE:FEED::/96. The calculation formula depends on a used prefix length and it is described in RFC 6052.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The whole IPv6 address is 2010:CAFE:FEED::D201:101 and it is added to Lo4 to simulate a subscriber.

```
CEL7(config)#interface Loopback4
CEL7(config-if)# ipv6 address 2010:CAFE:FEED::D201:101/128
CEL7#sh ipv6 route vrf csc-cust ospf
IPv6 Routing Table - csc-cust - 9 entries
[..]
OE2 64:FF9B::/96 [110/20]
     via FE80::21E:49FF:FE1C:A200, Tunnel2
OE2 2001:B000::/32 [110/20]
     via FE80::21E:49FF:FE1C:A200, Tunnel2
OE2 2006:BEEF::/96 [110/20]
     via FE80::21E:49FF:FE1C:A200, Tunnel2
OE2 2007::/64 [110/20]
     via FE80::21E:49FF:FE1C:A200, Tunnel2
OE2 2010:CAFE:FEED::/96 [110/20]
     via FE80::21E:49FF:FE1C:A200, Tunnel2
```

Verification of the IP reachability. You can start a packet capture and stop it after a ping.

The destination 2007::000A:0243:0600:0000 embeds the IPv4 destination address which is 10.2.67.6. The "A:0243:06" part is a hexadecimal representation of 10.2.67.6.

```
CEL7#ping vrf csc-cust 2007::000A:0243:0600:0000 source 2010:CAFE:FEED::D201:0101
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2007::A:243:600:0, timeout is 2 seconds:
Packet sent with a source address of 2010:CAFE:FEED::D201:101%csc-cust
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 45/62/83 ms
```

The 210.1.1.1 address is embedded into the IPv6 source address as D202:0101.

```
CER6#show mon cap cap2 buffer brief
 ----------------------------------------------------------------------------
 #   size   timestamp     source              destination      dscp    protocol
 ----------------------------------------------------------------------------
   0   98     0.000000    210.1.1.1         ->  10.2.67.6        0  BE   ICMP
   1   98     0.102000    210.1.1.1         ->  10.2.67.6        0  BE   ICMP
   2   98     0.170997    210.1.1.1         ->  10.2.67.6        0  BE   ICMP
   3   98     0.250994    210.1.1.1         ->  10.2.67.6        0  BE   ICMP
   4   98     0.359982    210.1.1.1         ->  10.2.67.6        0  BE   ICMP
```

You can test what happens if you change the source IP.

```
CEL7(config)#int lo4
CEL7(config-if)#ipv6 address 2010:CAFE:FEED::D201:1A1/128
CER6#mon cap cap2 start
Started capture point : cap2
CEL7#ping vrf csc-cust 2007::000A:0243:0600:0000 source 2010:CAFE:FEED::D201:01A1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2007::A:243:600:0, timeout is 2 seconds:
Packet sent with a source address of 2010:CAFE:FEED::D201:1A1%csc-cust
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 67/84/101 ms
CER6#mon cap cap2 stop
Stopped capture point : cap2
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

There is a new IP source address.

```
CER6#show mon cap cap2 buffer brief
------------------------------------------------------------------------------
 #   size   timestamp     source              destination      dscp    protocol
 ------------------------------------------------------------------------------
   0   98     0.000000   210.1.1.1         ->  10.2.67.6        0  BE   ICMP
   1   98     0.102000   210.1.1.1         ->  10.2.67.6        0  BE   ICMP
   2   98     0.170997   210.1.1.1         ->  10.2.67.6        0  BE   ICMP
   3   98     0.250994   210.1.1.1         ->  10.2.67.6        0  BE   ICMP
   4   98     0.359982   210.1.1.1         ->  10.2.67.6        0  BE   ICMP
   5   98    59.126993   210.1.1.161       ->  10.2.67.6        0  BE   ICMP
   6   98    59.218998   210.1.1.161       ->  10.2.67.6        0  BE   ICMP
   7   98    59.300995   210.1.1.161       ->  10.2.67.6        0  BE   ICMP
   8   98    59.403986   210.1.1.161       ->  10.2.67.6        0  BE   ICMP
   9   98    59.474982   210.1.1.161       ->  10.2.67.6        0  BE   ICMP
```

No status of NAT64 translations though as this is a stateless form of NAT.

```
CER7#sh nat64 trans

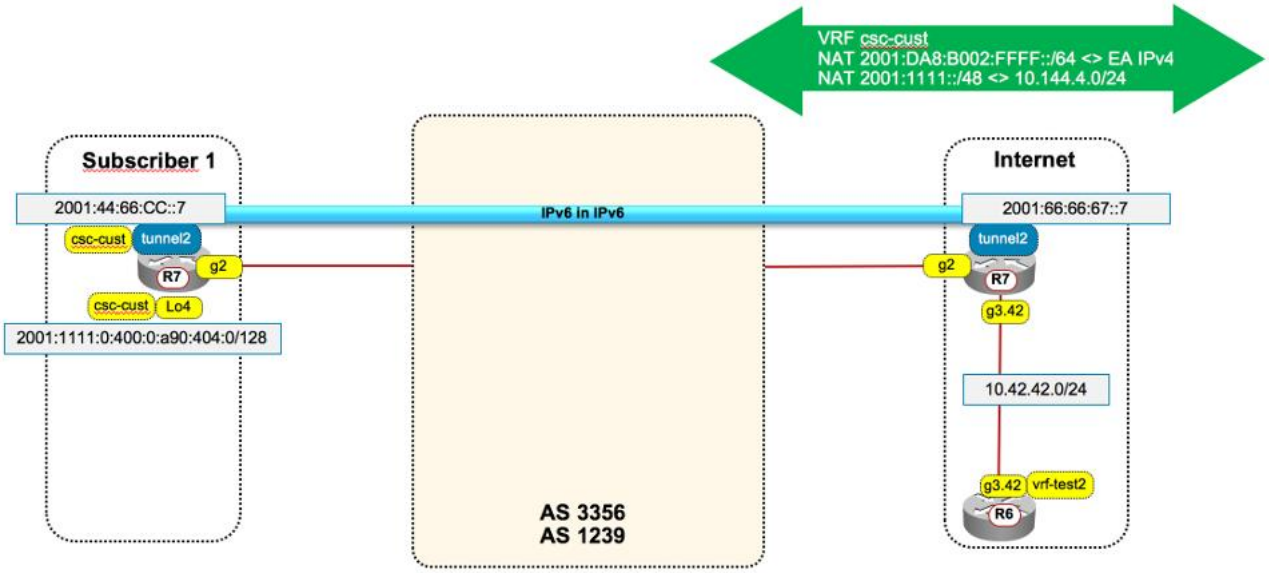
Proto  Original IPv4          Translated IPv4
       Translated IPv6        Original IPv6
------------------------------------------------------------------------------


Total number of translations: 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.6: MAP-T

Configure an another stateless method to translate IPv4 and IPv6 addresses. This is MAP-T which does not require any form of tunnelling. The IPv6 in IPv6 tunnel here is used to avoid IPv6 advertisement of Lo4 addresses into the upstream providers and to provide a leaking between VRF csc-cust and the global routing table.



Configure routers as follows. The default mapping rule is the stateless prefix assigned to the external end-points. The basic mapping rule prefixes are used by internal subscribers. The IPv6 prefix is the original source prefix and the IPv4 is the external pool used for translation.

CER7:

```
!
interface GigabitEthernet2
 description to SP2R2
 ip address 20.40.27.7 255.255.255.0
 negotiation auto
 nat64 enable
 ipv6 address FE80::7 link-local
 ipv6 address 2001:20:40:27::7/64
!
interface GigabitEthernet3.42
 encapsulation dot1Q 42
 ip address 10.42.42.1 255.255.255.0
 nat64 enable
 ipv6 enable
!
nat64 map-t domain 1
 default-mapping-rule 2001:DA8:B001:FFFF::/64
 basic-mapping-rule
  ipv6-prefix 2001:1111::/48
  ipv4-prefix 10.144.4.0/24
  port-parameters share-ratio 1
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

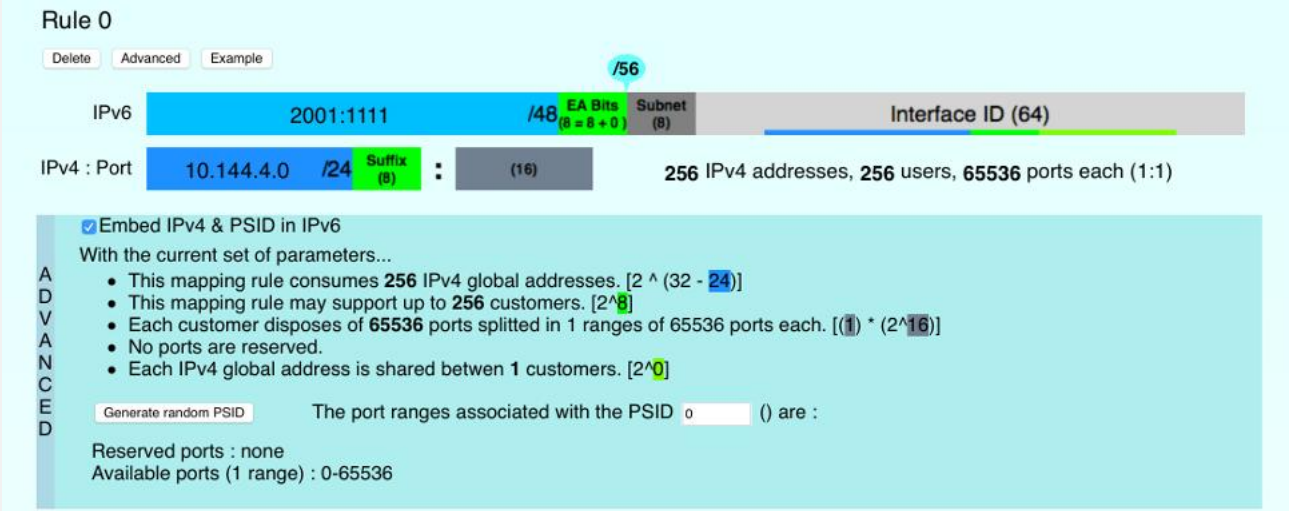
CER6:

```
!
ip vrf vrf-test2
 rd 2:2
!
interface GigabitEthernet3.42
 encapsulation dot1Q 42
 ip vrf forwarding vrf-test2
 ip address 10.42.42.4 255.255.255.0
!
ip route vrf vrf-test2 0.0.0.0 0.0.0.0 10.42.42.1
!
```

Set an additional IPv6 address on the Lo4 interface to simulate the 10.144.4.4 address.

```
CEL7(config)#int lo4
CEL7(config-if)#ipv6 add 2001:1111:0:400:0:a90:404:0/128
```

There is an algorithm behind the translation in both directions. The calculated IPv6 address depends of the number of planned subscribers and ports used by each of them. This can be planned in the MAP-T tool (http://6lab.cisco.com/map/MAP.php). A picture is depicted as follows.



Check the routing table to see the exit to the external network.

```
CEL7#sh ipv6 route b
IPv6 Routing Table - default - 18 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
       NDr - Redirect, RL - RPL, O - OSPF Intra, OI - OSPF Inter
       OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1
       ON2 - OSPF NSSA ext 2, la - LISP alt, lr - LISP site-registrations
       ld - LISP dyn-eid, lA - LISP away, a - Application
B   64:FF9B::/96 [20/0]
     via FE80::2, GigabitEthernet2
B   2001:33:56::3:31/128 [20/0]
     via FE80::2, GigabitEthernet2
```

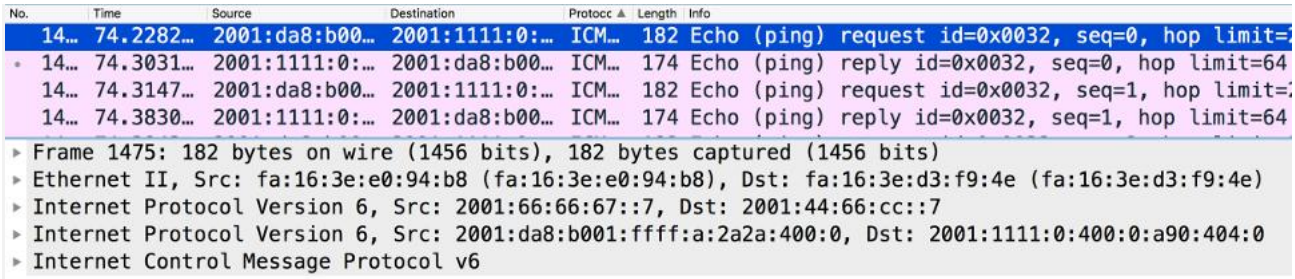

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
B    2001:33:56::4:41/128 [20/0]
      via FE80::2, GigabitEthernet2
B    2001:66:66:67::/64 [20/0]
      via FE80::2, GigabitEthernet2
B    2001:DA8:B001:FFFF::/64 [20/0]
      via FE80::2, GigabitEthernet2
B    2006:BEEF::/96 [20/0]
      via FE80::2, GigabitEthernet2
B    2007::/64 [20/0]
      via FE80::2, GigabitEthernet2
B    2010:CAFE:FEED::/96 [20/0]
      via FE80::2, GigabitEthernet2
```

Verify the IP reachability.

```
CER6#ping vrf vrf-test2 10.144.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.144.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 76/127/232 ms
```

A packet capture from the g2 interface of CER7.



You can check your translated IP address without a packet capture.

```
CER6#telnet 10.144.4.4 /vrf vrf-test2
Trying 10.144.4.4 ... Open


User Access Verification

Username: cisco
Password: cisco

CEL7#
CEL7#sh users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:02:50
*  1 vty 0     cisco      idle                 00:00:00
                                                  2001:DA8:B001:FFFF:A:2A2A:400:0

  Interface    User               Mode         Idle     Peer Address
```

Show MAP-T statistics.

```
CER7#sh nat64 statistics
NAT64 Statistics

Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Sessions found: 165
Sessions created: 20
Expired translations: 18
Global Stats:
   Packets translated (IPv4 -> IPv6)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

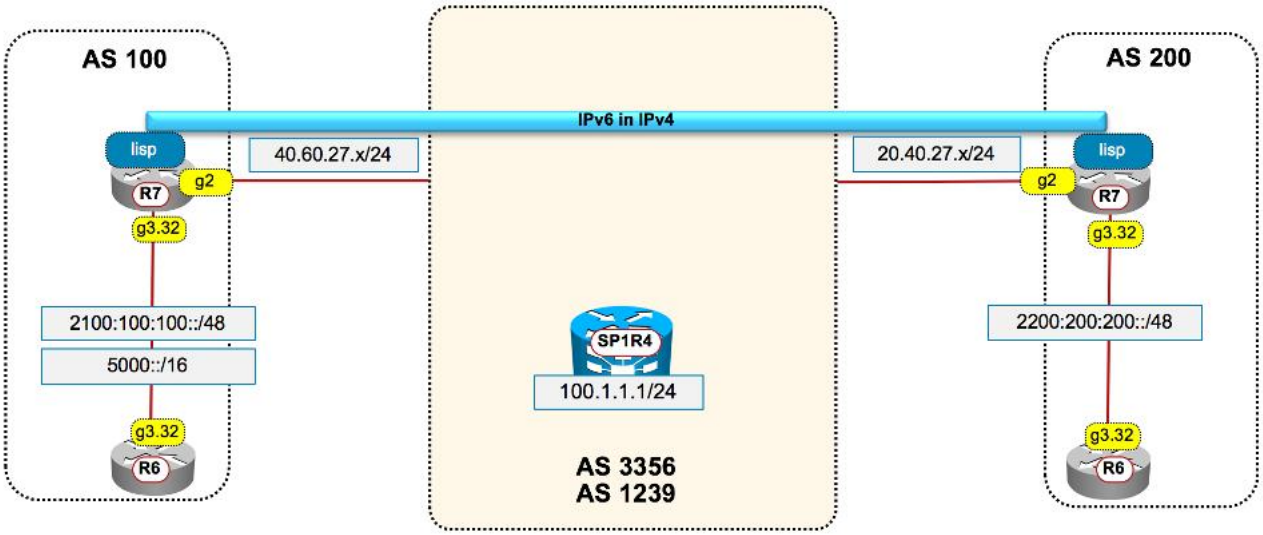
```
      Stateless: 35
      Stateful: 85
      MAP-T: 74
   Packets translated (IPv6 -> IPv4)
      Stateless: 45
      Stateful: 100
      MAP-T: 39

Interface Statistics
   GigabitEthernet2 (IPv4 configured, IPv6 configured):
      Packets translated (IPv4 -> IPv6)
         Stateless: 0
         Stateful: 0
         MAP-T: 0
      Packets translated (IPv6 -> IPv4)
         Stateless: 0
         Stateful: 0
         MAP-T: 0
      Packets dropped: 11
   GigabitEthernet3.32 (IPv4 configured, IPv6 configured):
      Packets translated (IPv4 -> IPv6)
         Stateless: 15
         Stateful: 50
         MAP-T: 0
      Packets translated (IPv6 -> IPv4)
         Stateless: 0
         Stateful: 0
         MAP-T: 0
      Packets dropped: 5
   GigabitEthernet3.42 (IPv4 configured, IPv6 configured):
      Packets translated (IPv4 -> IPv6)
         Stateless: 0
         Stateful: 0
         MAP-T: 74
      Packets translated (IPv6 -> IPv4)
         Stateless: 0
         Stateful: 0
         MAP-T: 0
      Packets dropped: 33
   Tunnel2 (IPv4 configured, IPv6 configured):
      Packets translated (IPv4 -> IPv6)
         Stateless: 0
         Stateful: 0
         MAP-T: 0
      Packets translated (IPv6 -> IPv4)
         Stateless: 45
         Stateful: 70
         MAP-T: 39
      Packets dropped: 40
Dynamic Mapping Statistics
   v6v4
      access-list nat64-acl pool pool1 refcount 0
         pool pool1:
            start 209.165.201.1 end 209.165.201.254
            total addresses 254, allocated 0 (0%)
            address exhaustion packet count 0
Limit Statistics
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.7: LISP

Configure the IPv6 in IPv4 tunnelling by using LISP (Locator-ID Separation Protocol).



The map server (MA) and map resolver (MR) role is played by the SP1R4 router. The RLOC (Routing Locator) are CEL7 and CER7 routers. The RLOC routers are registering their local prefixed on MA which is 100.1.1.1. The **locator-table lisp** command declares that RLOC are reachable through the routing table in VRF named lisp.

Configure devices as follows.

MS/MR (SP1R4):

```
!
vrf lisp
 address-family ipv4 unicast
  import route-target
   1:100
  !
  export route-target
   1:100
  !
 !
!
interface Loopback100
 vrf lisp
 ipv4 address 100.1.1.1 255.255.255.255
!
router bgp 3356
 vrf lisp
  rd 1:100
  address-family ipv4 unicast
   redistribute connected
  !
 !
!
router lisp
 address-family ipv6 unicast
  map-resolver
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  map-server
 !
 site as200
  eid-prefix 2200:200::/32 accept-more-specifics
  allowed-locator 20.40.27.7
  authentication-key encrypted 00050054540B
 !
 site as100
  eid-prefix 2100:100::/32 accept-more-specifics
  eid-prefix 5000::/16 accept-more-specifics
  allowed-locator 40.60.27.7
  authentication-key encrypted 050A155E711C
 !
 locator-table lisp
!
```

CEL7:

```
!
router lisp
 eid-table default instance-id 0
  database-mapping 2100:100:100::/48 40.60.27.7 priority 1 weight 10
  database-mapping 2100:100:100::10/128 40.60.27.7 priority 1 weight 10
  database-mapping 5000::/16 40.60.27.7 priority 100 weight 20
  exit
 !
 ipv6 itr map-resolver 100.1.1.1
 ipv6 itr
 ipv6 etr map-server 100.1.1.1 key as100
 ipv6 etr
 exit
!
```

CER7 has a static mapping entry which does not require the MA/MR server.

CER7:

```
!
router lisp
 eid-table default instance-id 0
  database-mapping 2200:200:200::/48 20.40.27.7 priority 1 weight 10
  map-cache 2100:100:100::10/128 40.60.27.7 priority 1 weight 60
  exit
 !
 ipv6 itr map-resolver 100.1.1.1
 ipv6 itr
 ipv6 etr map-server 100.1.1.1 key as200
 ipv6 etr
 exit
!
```

Verify reachability to MA/MR.

```
CEL7#sh ip route 100.1.1.1
Routing entry for 100.1.1.1/32
  Known via "ospf 100", distance 110, metric 1
  Tag Complete, Path Length == 1, AS 3356, , type extern 2, forward metric 1
  Last update from 40.60.27.2 on GigabitEthernet2, 00:13:07 ago
  Routing Descriptor Blocks:
  * 40.60.27.2, from 33.56.2.2, 00:13:07 ago, via GigabitEthernet2
      Route metric is 1, traffic share count is 1
      Route tag 3489664284

CER7#sh ip route 100.1.1.1
Routing entry for 100.1.1.1/32
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Known via "bgp 200", distance 20, metric 0
  Tag 1239, type external
  Last update from 20.40.27.2 00:13:31 ago
  Routing Descriptor Blocks:
  * 20.40.27.2, from 20.40.27.2, 00:13:31 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2
      Route tag 1239
      MPLS label: none
```

Check reachability with the source address used for prefix registrations.

```
CEL7#ping 100.1.1.1 sou 40.60.27.7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 40.60.27.7
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 29/41/81 ms

CER7#ping 100.1.1.1 sou 20.40.27.7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 20.40.27.7
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 38/94/191 ms
```

Verification of the control-plane, RLOC status and EID (Endpoint Identifier) prefix registrations.

```
RP/0/0/CPU0:SP1R4#sh lisp site
LISP Site Registration Information
* = Some locators are down or unreachable

Site Name      Last      Up   Who Last             Inst     EID Prefix
               Register       Registered           ID
as100          never     no   --                            2100:100::/32
               00:00:16  yes  40.60.27.7                    2100:100:100::/48
               00:00:16  yes  40.60.27.7                    2100:100:100::10/128
               00:00:16  yes  40.60.27.7                    5000::/16
as200          never     no   --                            2200:200::/32
               00:00:23  yes  20.40.27.7                    2200:200:200::/48
```

If the RLOC status is admin-down it means that the RLOC does not have an active prefix locally accessible.

```
RP/0/0/CPU0:SP1R4#sh lisp site det
LISP Site Registration Information

Site name: as100
Allowed configured locators:
  40.60.27.7
Allowed EID-prefixes:
  EID-prefix: 2100:100::/32
    First registered:     never
    Routing table tag:    0
    Origin:               Configuration, accepting more specifics
    Merge active:         No
    Proxy reply:          No
    TTL:                  00:00:00.000
    State:                unknown
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    No registrations.
  EID-prefix: 2100:100:100::/48
    First registered:     18:15:40
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    Routing table tag:    0
    Origin:               Dynamic, more specific of 2100:100::/32
    Merge active:         No
    Proxy reply:          No
    TTL:                  1d00h
    State:                complete
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    ETR 40.60.27.7, last registered 00:00:32, no proxy-reply, map-notify
                    TTL 1d00h, no merge, hash-function sha1, nonce 0x1afa7bf2-0x614dbd0d
                    state complete, no security-capability
                    xTR-ID 0x7ded4a6e-0x9ea53d7a-0xdd082132-0xbf8c1d50
                    site-ID unspecified
      Locator     Local  State      Pri/Wgt  Scope
      40.60.27.7  yes    up           1/10   IPv4 none
  EID-prefix: 2100:100:100::10/128
    First registered:     18:15:40
    Routing table tag:    0
    Origin:               Dynamic, more specific of 2100:100::/32
    Merge active:         No
    Proxy reply:          No
    TTL:                  1d00h
    State:                complete
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    ETR 40.60.27.7, last registered 00:00:32, no proxy-reply, map-notify
                    TTL 1d00h, no merge, hash-function sha1, nonce 0x1afa7bf2-0x614dbd0d
                    state complete, no security-capability
                    xTR-ID 0x7ded4a6e-0x9ea53d7a-0xdd082132-0xbf8c1d50
                    site-ID unspecified
      Locator     Local  State      Pri/Wgt  Scope
      40.60.27.7  yes    up           1/10   IPv4 none
  EID-prefix: 5000::/16
    First registered:     18:17:32
    Routing table tag:    0
    Origin:               Configuration, accepting more specifics
    Merge active:         No
    Proxy reply:          No
    TTL:                  1d00h
    State:                complete
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    ETR 40.60.27.7, last registered 00:00:32, no proxy-reply, map-notify
                    TTL 1d00h, no merge, hash-function sha1, nonce 0x1afa7bf2-0x614dbd0d
                    state complete, no security-capability
                    xTR-ID 0x7ded4a6e-0x9ea53d7a-0xdd082132-0xbf8c1d50
                    site-ID unspecified
      Locator     Local  State      Pri/Wgt  Scope
      40.60.27.7  yes    up         100/20   IPv4 none
Site name: as200
Allowed configured locators:
  20.40.27.7
Allowed EID-prefixes:
  EID-prefix: 2200:200::/32
    First registered:     never
    Routing table tag:    0
    Origin:               Configuration, accepting more specifics
    Merge active:         No
    Proxy reply:          No
    TTL:                  00:00:00.000
    State:                unknown
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    No registrations.
  EID-prefix: 2200:200:200::/48
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    First registered:     18:18:13
    Routing table tag:    0
    Origin:               Dynamic, more specific of 2200:200::/32
    Merge active:         No
    Proxy reply:          No
    TTL:                  1d00h
    State:                complete
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    ETR 20.40.27.7, last registered 00:00:39, no proxy-reply, map-notify
                    TTL 1d00h, no merge, hash-function sha1, nonce 0x747ec3ea-0xb1fb258b
                    state complete, no security-capability
                    xTR-ID 0x785181d2-0x31d206e7-0x7178f095-0xca7ef95f
                    site-ID unspecified
      Locator     Local  State      Pri/Wgt  Scope
      20.40.27.7  yes    up           1/10   IPv4 none
```

Check the routing. There are no destination prefixes in RIB so the LISP solution will be triggered.

```
CEL7#sh ipv6 route | i 2100:
C   2100:100:100::/48 [0/0]
L   2100:100:100::1/128 [0/0]

CEL7#sh ipv6 route 2200:200:200::1
% Route not found
```

Check the current LISP database of locally reachable IPv6 prefixes.

```
CEL7#sh ipv6 lisp database
LISP ETR IPv6 Mapping Database for EID-table default (IID 0), LSBs: 0x1
Entries total 3, no-route 0, inactive 0

2100:100:100::/48
  Locator     Pri/Wgt  Source     State
  40.60.27.7    1/10   cfg-addr   site-self, reachable
2100:100:100::10/128
  Locator     Pri/Wgt  Source     State
  40.60.27.7    1/10   cfg-addr   site-self, reachable
5000::/16
  Locator     Pri/Wgt  Source     State
  40.60.27.7  100/20   cfg-addr   site-self, reachable
```

In the LISP cache there is a default route only so no more detailed information where to go to reach the destination.

```
CEL7#sh ipv6 lisp map-cache
LISP IPv6 Mapping Cache for EID-table default (IID 0), 1 entries

::/0, uptime: 00:00:08, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
```

The LISP database on CER7.

```
CER7#sh ipv6 lisp database
LISP ETR IPv6 Mapping Database for EID-table default (IID 0), LSBs: 0x1
Entries total 1, no-route 0, inactive 0

2200:200:200::/48
  Locator     Pri/Wgt  Source     State
  20.40.27.7    1/10   cfg-addr   site-self, reachable
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

In the LISP cache there is also a configured static entry.

```
CER7#sh ipv6 lisp map-cache
LISP IPv6 Mapping Cache for EID-table default (IID 0), 2 entries

::/0, uptime: 00:00:07, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
2100:100:100::10/128, uptime: 00:00:07, expires: never, via static
  Locator      Uptime    State      Pri/Wgt
  40.60.27.7   00:00:07  up           1/60
```

Test IP connectivity.

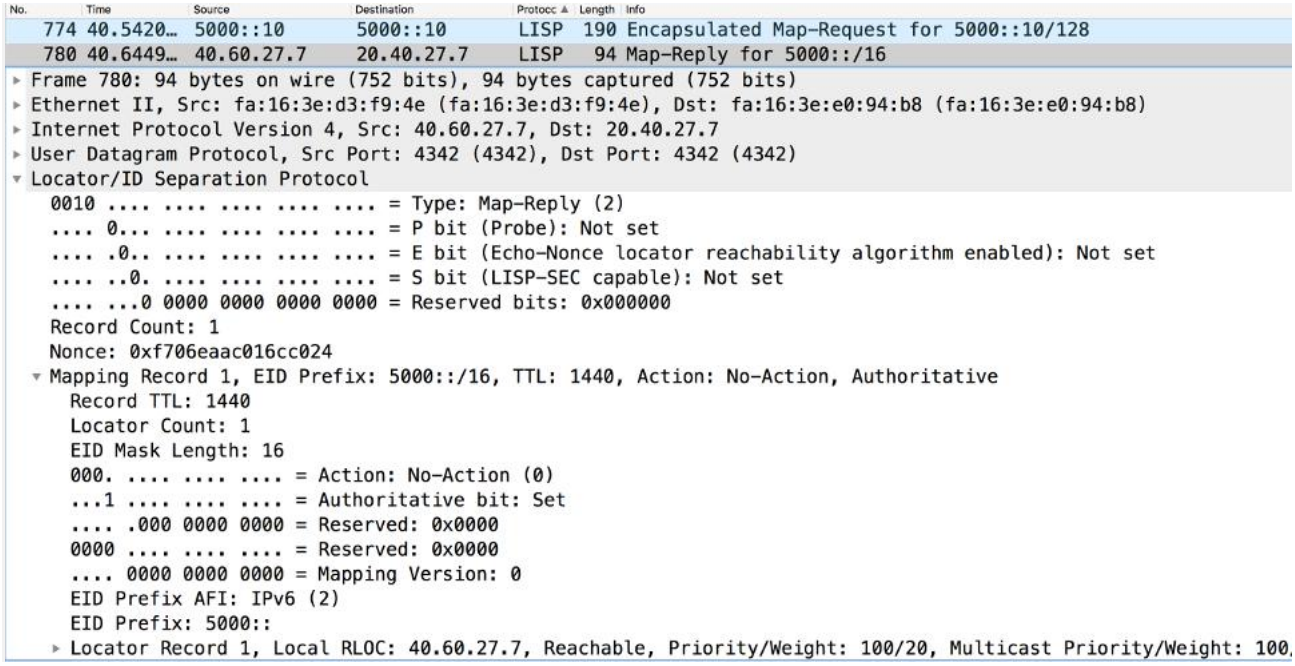
```
CER6#ping 5000::10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5000::10, timeout is 2 seconds:
..!!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 59/88/119 ms
```

In a picture depicted below you can see a map request which is sent by RLOC CER7 to the map resolver (MR) asking where the address 5000::10 is currently located.

```
No.  Time      Source        Destination   Protocc ▲ Length Info
  774 40.5420… 5000::10      5000::10      LISP      190 Encapsulated Map-Request for 5000::10/128
  780 40.6449… 40.60.27.7    20.40.27.7    LISP       94 Map-Reply for 5000::/16
▸ Frame 774: 190 bytes on wire (1520 bits), 190 bytes captured (1520 bits)
▸ Ethernet II, Src: fa:16:3e:e0:94:b8 (fa:16:3e:e0:94:b8), Dst: fa:16:3e:d3:f9:4e (fa:16:3e:d3:f9:4e)
▸ Internet Protocol Version 4, Src: 20.40.27.7, Dst: 100.1.1.1
▸ User Datagram Protocol, Src Port: 4342 (4342), Dst Port: 4342 (4342)
▸ Locator/ID Separation Protocol
▸ Internet Protocol Version 6, Src: 5000::10, Dst: 5000::10
▸ User Datagram Protocol, Src Port: 4342 (4342), Dst Port: 4342 (4342)
▾ Locator/ID Separation Protocol
    0001 .... .... .... .... .... = Type: Map-Request (1)
  ▸ .... 0100 00.. .... .... .... = Flags: 0x000010
    .... .... ..00 0000 000. .... = Reserved bits: 0x000000
    .... .... .... .... ...0 0000 = ITR-RLOC Count: 0
    Record Count: 1
    Nonce: 0xf706eaac016cc024
    Source EID AFI: IPv6 (2)
    Source EID: 2200:200:200::20
  ▸ ITR-RLOC 1: 20.40.27.7
  ▸ Map-Request Record 1: 5000::10/128
  ▸ Map-Reply Record
```

The request is forwarded by MR to the destination RLOC CEL7 which responds to CER7 as follows.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński



This response contains information needed to encapsulate the IPv6 packet into the IPv4 header. The IPv4 destination address of the external header is RLOC 40.60.27.7.

The map cache entries are updated.

```
CEL7#sh ipv6 lisp map-cache
LISP IPv6 Mapping Cache for EID-table default (IID 0), 2 entries

::/0, uptime: 00:06:43, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
2200:200:200::/48, uptime: 00:00:55, expires: 23:59:04, via map-reply, complete
  Locator      Uptime    State      Pri/Wgt
  20.40.27.7   00:00:55  up           1/10


CER7#sh ipv6 lisp map-cache
LISP IPv6 Mapping Cache for EID-table default (IID 0), 3 entries

::/0, uptime: 00:04:47, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
2100:100:100::10/128, uptime: 00:04:47, expires: never, via static
  Locator      Uptime    State      Pri/Wgt
  40.60.27.7   00:04:47  up           1/60
5000::/16, uptime: 00:01:16, expires: 23:58:43, via map-reply, complete
  Locator      Uptime    State      Pri/Wgt
  40.60.27.7   00:01:16  up         100/20
```

The LISP forwarding table.

```
CEL7#sh ipv6 lisp forwarding eid remote 2200:200:200::20
Prefix                 Fwd action  Locator status bits
2200:200:200::/48      encap       0x00000001
  packets/bytes       8/944
  path list 7FCA109A4A58, 3 locks, per-destination, flags 0x49 [shble, rif, hwcn]
    ifnums:
      LISP0(37): 20.40.27.7
    1 path
      path 7FCA10A87E00, share 10/10, type attached nexthop, for IPv6
        nexthop 20.40.27.7 LISP0, IPV6 midchain out of LISP0, addr 20.40.27.7
7FCA102E73A8
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    1 output chain
      chain[0]: IPV6 midchain out of LISP0, addr 20.40.27.7 7FCA102E73A8
                IP adj out of GigabitEthernet2, addr 40.60.27.2 7FC9E3A1F898


CER7#sh ipv6 lisp forwarding eid re 5000::10
Prefix                 Fwd action  Locator status bits
5000::/16              encap       0x00000001
  packets/bytes       9/1062
  path list 7F81907A9990, 3 locks, per-destination, flags 0x49 [shble, rif, hwcn]
    ifnums:
      LISP0(34): 40.60.27.7
    1 path
      path 7F81BB620810, share 20/20, type attached nexthop, for IPv6
        nexthop 40.60.27.7 LISP0, IPV6 midchain out of LISP0, addr 40.60.27.7
7F8190508128
    1 output chain
      chain[0]: IPV6 midchain out of LISP0, addr 40.60.27.7 7F8190508128
                IP adj out of GigabitEthernet2, addr 20.40.27.2 7F818FC582A8
```

Verify the IPv6 reachability over the IPv4 network.

```
CER6#ping 5000::10 source 2200:200:200::20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5000::10, timeout is 2 seconds:
Packet sent with a source address of 2200:200:200::20
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/90/135 ms
```

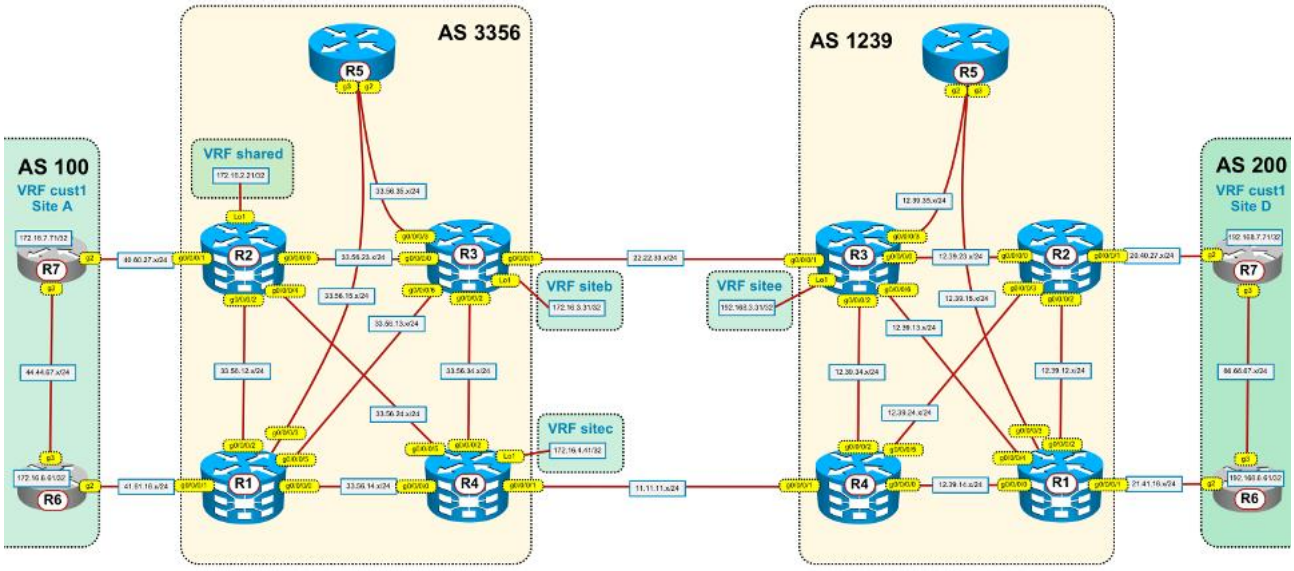
No.	Time	Source	Destination	Protocc	Length	Info
514	26.9635…	fe80::2	fe80::7	ICM…	86	Neighbor Advertisement fe80::2 (rtr, sol, ovr) is a
811	42.3978…	2200:200:200…	5000::10	ICM…	150	Echo (ping) request id=0x186e, seq=1, hop limit=63
812	42.4425…	5000::10	2200:200:200…	ICM…	150	Echo (ping) reply id=0x186e, seq=1, hop limit=63 (r
813	42.4512…	2200:200:200…	5000::10	ICM…	150	Echo (ping) request id=0x186e, seq=2, hop limit=63
814	42.5079…	5000::10	2200:200:200…	ICM…	150	Echo (ping) reply id=0x186e, seq=2, hop limit=63 (r
815	42.5268…	2200:200:200…	5000::10	ICM…	150	Echo (ping) request id=0x186e, seq=3, hop limit=63
817	42.6040…	5000::10	2200:200:200…	ICM…	150	Echo (ping) reply id=0x186e, seq=3, hop limit=63 (r

```
▸ Frame 812: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits)
▸ Ethernet II, Src: fa:16:3e:d3:f9:4e (fa:16:3e:d3:f9:4e), Dst: fa:16:3e:e0:94:b8 (fa:16:3e:e0:94:b8)
▸ Internet Protocol Version 4, Src: 40.60.27.7, Dst: 20.40.27.7
▸ User Datagram Protocol, Src Port: 1090 (1090), Dst Port: 4341 (4341)
▸ Locator/ID Separation Protocol (Data)
▸ Internet Protocol Version 6, Src: 5000::10, Dst: 2200:200:200::20
▸ Internet Control Message Protocol v6
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 16: IP Multicast

The following picture depicts a physical topology used in the multicast section.



This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch16.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch16.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch16.unl`

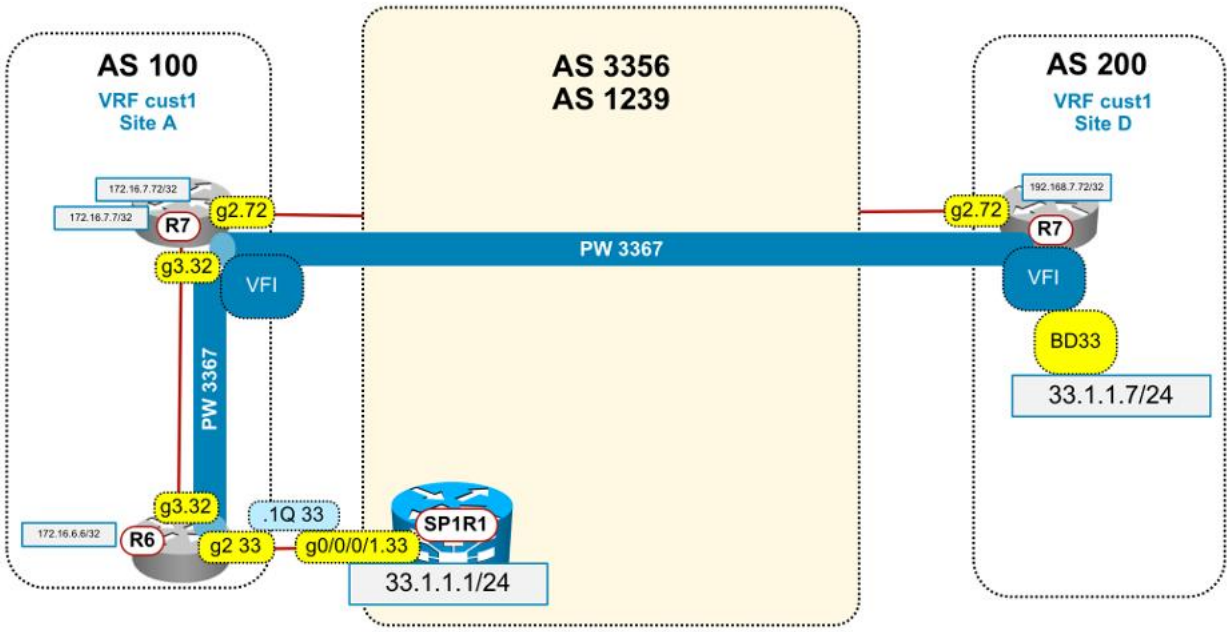
© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: Native Multicast

In this task you will configure the native multicast.

Task 1.1: PIM ASM Static RP

Enable the Any Source Multicast (ASM) multicast transmission over the L2 network. To simulate LAN you can use a PW 3367 configured in the previous L2VPN section.



Verify that pseudowires are UP. If not, please troubleshoot and resolve potential issues.

```
CEL7#sh mpls l2 vc 3367

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
VFI hvpls      vfi                        172.16.6.6      3367       UP
VFI hvpls      vfi                        192.168.7.72    3367       UP
```

Enable multicast in a form of an OSPF session.

```
RP/0/0/CPU0:SP1R1(config)#router ospf AS3356
RP/0/0/CPU0:SP1R1(config-ospf)#area 0
RP/0/0/CPU0:SP1R1(config-ospf-ar)#int g0/0/0/1.33

CER7(config)#int bdi33
CER7(config-if)#ip ospf 2 area 0
```

Verify the OSPF session.

```
CER7#sh ip ospf nei

Neighbor ID     Pri   State           Dead Time   Address         Interface
33.56.1.1         1   FULL/BDR        00:00:32    33.1.1.1        BDI33
17.1.1.1          0   FULL/  -        00:00:32    22.1.1.1        Tunnel2
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

How to check which multicast groups are joined on an interface?

```
CER7#sh ip int bdi33
BDI33 is up, line protocol is up
  Internet address is 33.1.1.7/24
  Broadcast address is 255.255.255.255
  Address determined by configuration file
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.2 224.0.0.1 224.0.0.22 224.0.0.13
      224.0.0.5 224.0.0.6
```

You can also enable devices to send and to receive the regular multicast traffic. To do this you need to turn PIM on router's interfaces. This is not required on receivers like set-top boxes (STB).

```
RP/0/0/CPU0:SP1R1(config)#multicast-routing
RP/0/0/CPU0:SP1R1(config-mcast)#add ipv4
RP/0/0/CPU0:SP1R1(config-mcast-default-ipv4)#int g0/0/0/1.33
RP/0/0/CPU0:SP1R1(config-mcast-default-ipv4-if)#enable

CER7(config)#ip multicast-routing distributed
CER7(config)#int bdi33
CER7(config-if)#ip pim sparse-mode
*May 25 19:18:09.048: %PIM-5-DRCHG: DR change from neighbor 0.0.0.0 to 33.1.1.7 on
interface BDI33
CER7(config-if)#ip igmp join 233.1.1.1
```

You can see that CER7 joined the mcast group 233.1.1.1.

```
CER7#sh ip igmp groups
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter   Group
Accounted
233.1.1.1        BDI33                    00:22:54  00:02:37  33.1.1.7
224.0.1.40       BDI33                    00:24:31  00:02:32  33.1.1.7
```

As a receiver CER7 should have the RPF neighbor. Otherwise an PIM join packet won't be sent to an RP or a source of multicast.

```
CER7#sh ip mroute 233.1.1.1
[..]
(*, 233.1.1.1), 00:24:02/00:02:28, RP 0.0.0.0, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    BDI33, Forward/Sparse, 00:24:02/00:02:28
```

Multicast is dropped.

```
RP/0/0/CPU0:SP1R1#ping 233.1.1.1 sou 33.1.1.1 count 2
Thu May 25 19:50:13.208 UTC
Type escape sequence to abort.
Sending 2, 100-byte ICMP Echos to 233.1.1.1, timeout is 2 seconds:
..

CER7#sh ip mroute count
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)
Group: 233.1.1.1, Source count: 0, Packets forwarded: 0, Packets received: 27
  RP-tree: Forwarding: 0/0/0/0, Other: 27/27/0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Add the RP address on CER7. R1 does not have to know that he plays the RP role.

```
CER7(config)#ip pim rp-address 33.1.1.1
```

Verify the multicast delivery.

```
RP/0/0/CPU0:SP1R1#ping 233.1.1.1 sou 33.1.1.1 count 2
Type escape sequence to abort.
Sending 2, 100-byte ICMP Echos to 233.1.1.1, timeout is 2 seconds:

Reply to request 0 from 33.1.1.7, 59 ms
Reply to request 1 from 33.1.1.7, 9 ms

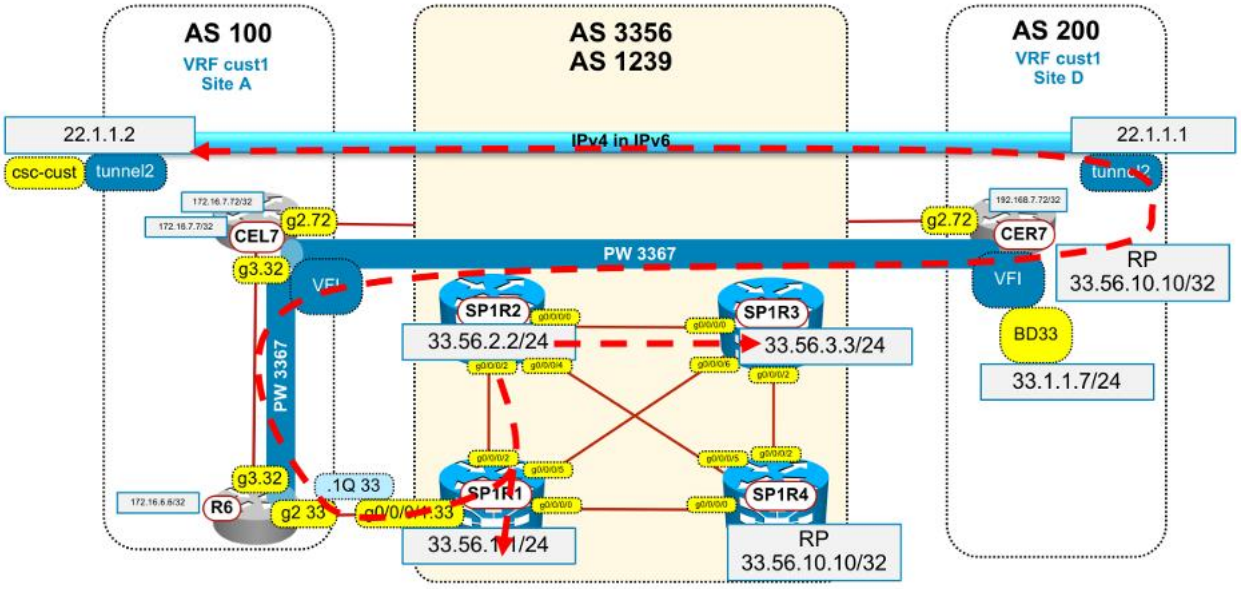
CER7#sh ip mroute 233.1.1.1
[..]
(*, 233.1.1.1), 01:40:22/stopped, RP 33.1.1.1, flags: SJPCLF
  Incoming interface: BDI33, RPF nbr 33.1.1.1
  Outgoing interface list: Null

(33.1.1.1, 233.1.1.1), 00:01:30/00:01:29, flags: PLFT
  Incoming interface: BDI33, RPF nbr 33.1.1.1
  Outgoing interface list: Null
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: PIM ASM with Static Anycast RP

Configure PIM sparse-mode in the AS 3356. Set R4 and CER7 as the static anycast RP for the group 233.33.56.2. Configure receivers on R1, R3 and CEL7.



Add PIM to interfaces and enable the multicast on all 4 SP1 routers based on the following example.

R2:

```
!
router pim
 address-family ipv4
  interface Loopback0
   enable
  !
  interface GigabitEthernet0/0/0/0
  interface GigabitEthernet0/0/0/2
  interface GigabitEthernet0/0/0/4
  !
multicast-routing
 address-family ipv4
  interface Loopback0
   enable
  !
  interface GigabitEthernet0/0/0/0
   enable
  !
  interface GigabitEthernet0/0/0/2
   enable
  !
  interface GigabitEthernet0/0/0/4
   enable
  !
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Each router should have 3 PIM neighbors from SP1.

```
RP/0/0/CPU0:SP1R2#sh pim nei
PIM neighbors in VRF default
Flag: B - Bidir capable, P - Proxy capable, DR - Designated Router,
      E - ECMP Redirect capable
      * indicates the neighbor created for this router

Neighbor Address              Interface              Uptime    Expires  DR pri   Flags

33.56.23.2*                   GigabitEthernet0/0/0/0 00:05:40  00:01:20 1       B E
33.56.23.3                    GigabitEthernet0/0/0/0 00:01:13  00:01:40 1 (DR) B
33.56.12.1                    GigabitEthernet0/0/0/2 00:00:44  00:01:31 1       B
33.56.12.2*                   GigabitEthernet0/0/0/2 00:05:40  00:01:21 1 (DR) B E
33.56.2.2*                    Loopback0              00:05:40  00:01:15 1 (DR) B E
33.56.24.2*                   GigabitEthernet0/0/0/4 00:05:40  00:01:42 1       B E
33.56.24.4                    GigabitEthernet0/0/0/4 00:00:06  00:01:43 1 (DR) B
```

Add a static RP configuration on each of the SP1 routers.

```
!
router pim
 address-family ipv4
  rp-address 33.56.10.10
!
```

Verify RP mappings. You can see that the whole multicast range is assigned by default to the RP 33.56.10.10.

```
RRP/0/0/CPU0:SP1R2#sh pim rp mapping
PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 33.56.10.10 (?), v2
    Info source: 0.0.0.0 (?), elected via config
      Uptime: 00:01:17, expires: never
```

Where is the RP address? Let's configure it.

```
RP/0/0/CPU0:SP1R4(config)#int lo10
RP/0/0/CPU0:SP1R4(config-if)#ip add 33.56.10.10/32
RP/0/0/CPU0:SP1R4(config-if)#router ospf AS3356
RP/0/0/CPU0:SP1R4(config-ospf)#area 0
RP/0/0/CPU0:SP1R4(config-ospf-ar)#int lo10
```

The RP in reachable via g0/0/0/4 interface which is as expected.

```
RP/0/0/CPU0:SP1R2#sh route 33.56.10.10
Routing entry for 33.56.10.10/32
  Known via "ospf AS3356", distance 110, metric 2, type intra area
  Installed May 27 09:26:24.277 for 00:00:41
  Routing Descriptor Blocks
    33.56.24.4, from 33.56.4.4, via GigabitEthernet0/0/0/4
      Route metric is 2
```

No multicast entries yet.

```
RP/0/0/CPU0:SP1R4#sh mrib route 233.33.56.2
No matching routes in MRIB route-DB
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Join the mcast stream on R1 and R3.

```
!
router igmp
 interface Loopback0
  join-group 233.33.56.2
!
```

After joining the group, the Outgoing Interface List (OIL) is no longer empty. There are 2 interfaces in OIL from where R4 received PIM Join messages. There is no incoming interface as the multicast stream is not transmitted yet.

```
RP/0/0/CPU0:SP1R4#sh mrib route 233.33.56.2
[..]
(*,233.33.56.2) RPF nbr: 0.0.0.0 Flags: C RPF
  Up: 00:00:09
  Outgoing Interface List
    GigabitEthernet0/0/0/0 Flags: F NS, Up: 00:00:05
    GigabitEthernet0/0/0/2 Flags: F NS, Up: 00:00:09
```

Send the traffic and check the mrib table.

```
RP/0/0/CPU0:SP1R2#ping 233.33.56.2 sou 33.56.2.2 c 2
Type escape sequence to abort.
Sending 2, 100-byte ICMP Echos to 233.33.56.2, timeout is 2 seconds:
..

RP/0/0/CPU0:SP1R4#sh mrib route 233.33.56.2
[..]
(*,233.33.56.2) RPF nbr: 0.0.0.0 Flags: C RPF
  Up: 00:03:14
  Outgoing Interface List
    GigabitEthernet0/0/0/0 Flags: F NS, Up: 00:03:10
    GigabitEthernet0/0/0/2 Flags: F NS, Up: 00:03:14
```

There is no incoming traffic. Enable the multicast routing on the interface Lo10 which is the RP's address. At this moment PIM is off on Lo10.

```
RP/0/0/CPU0:SP1R4#sh pim int
[..]
PIM interfaces in VRF default
Address               Interface                     PIM  Nbr    Hello  DR     DR
                                                         Count  Intvl  Prior

33.56.4.4             Loopback0                     on   1      30     1      this system
33.56.10.10           Loopback10                    off  0      30     1      not elected

RP/0/0/CPU0:SP1R4(config)#multicast-routing
RP/0/0/CPU0:SP1R4(config-mcast)#add ipv4
RP/0/0/CPU0:SP1R4(config-mcast-default-ipv4)#int lo10
RP/0/0/CPU0:SP1R4(config-mcast-default-ipv4-if)#enable
```

Seems it still does not work.

```
RP/0/0/CPU0:SP1R2#ping 233.33.56.2 sou 33.56.2.2 c 2
Type escape sequence to abort.
Sending 2, 100-byte ICMP Echos to 233.33.56.2, timeout is 2 seconds:
..
```

There is a tunnel which the PIM register message encapsulated in the unicast header sent from the multicast source to the RP. This means the source is trying to register the multicast.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R4#sh mrib route 233.33.56.2
[..]
(*,233.33.56.2) RPF nbr: 33.56.10.10 Flags: C RPF
  Up: 00:10:08
  Incoming Interface List
    Decapstunnel0 Flags: A, Up: 00:00:42
  Outgoing Interface List
    GigabitEthernet0/0/0/0 Flags: F NS, Up: 00:10:03
    GigabitEthernet0/0/0/2 Flags: F NS, Up: 00:10:08

(33.56.2.2,233.33.56.2) RPF nbr: 33.56.24.2 Flags: L C RPF
  Up: 00:00:22
  Incoming Interface List
    Decapstunnel0 Flags: A, Up: 00:00:22
  Outgoing Interface List
    GigabitEthernet0/0/0/0 Flags: F NS, Up: 00:00:22
    GigabitEthernet0/0/0/2 Flags: F NS, Up: 00:00:22
    GigabitEthernet0/0/0/5 Flags: NS, Up: 00:00:22
```

Wait a second or send more packets. The RP completes the registration and switches over the multicast tree from RPT (Rendezvous Point Tree, also known as Shared Tree) to SPT (Shortest Path Tree).

```
RP/0/0/CPU0:SP1R2#ping 233.33.56.2 sou 33.56.2.2 c 4
Type escape sequence to abort.
Sending 4, 100-byte ICMP Echos to 233.33.56.2, timeout is 2 seconds:

Reply to request 0 from 33.56.1.1, 9 ms
Reply to request 0 from 33.56.3.3, 9 ms
Reply to request 1 from 33.56.3.3, 9 ms
Reply to request 1 from 33.56.1.1, 9 ms
Reply to request 2 from 33.56.1.1, 1 ms
Reply to request 2 from 33.56.3.3, 9 ms
Reply to request 3 from 33.56.3.3, 1 ms
Reply to request 3 from 33.56.1.1, 1 ms
```

The multicast tree is now switched to SPT.

```
RP/0/0/CPU0:SP1R4#sh mrib route 233.33.56.2
[..]
(*,233.33.56.2) RPF nbr: 33.56.10.10 Flags: C RPF
  Up: 00:11:12
  Incoming Interface List
    Decapstunnel0 Flags: A, Up: 00:01:46
  Outgoing Interface List
    GigabitEthernet0/0/0/0 Flags: F NS, Up: 00:11:07
    GigabitEthernet0/0/0/2 Flags: F NS, Up: 00:11:12

(33.56.2.2,233.33.56.2) RPF nbr: 33.56.10.10 Flags: L RPF
  Up: 00:01:26
  Incoming Interface List
    Decapstunnel0 Flags: A, Up: 00:00:11
```

Now, configure the other part of the network.

```
CER7(config)#ip pim rp-address 33.56.10.10
CER7(config)#int t2
CER7(config-if)#ip pim sparse-mode

CEL7(config)#ip multicast-routing distributed
CEL7(config)#ip multicast-routing vrf csc-cust distributed
CEL7(config)#int t2
CEL7(config-if)#ip pim sparse-mode
CEL7(config-if)#ip igmp join-group 233.33.56.2
CEL7(config-if)#ip pim vrf csc-cust rp-address 33.56.10.10
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The incoming interface is Null as no registration is done on CER7. The nearest RP to the multicast source is R4 and all registrations are done there.

```
CER7#sh ip mroute 233.33.56.2
[..]
(*, 233.33.56.2), 04:33:05/00:02:56, RP 33.56.10.10, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Tunnel2, Forward/Sparse, 04:33:05/00:02:56
```

To make RP on CER7 aware of the active source and provide resiliency of RP itself you can configure MSDP.

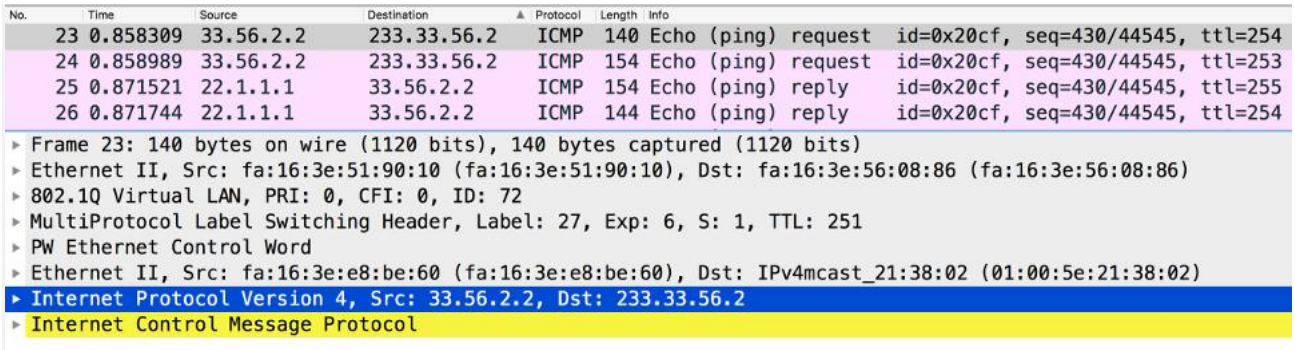
```
RP/0/0/CPU0:SP1R4(config)#router msdp
RP/0/0/CPU0:SP1R4(config-msdp)#connect-source Loopback0
RP/0/0/CPU0:SP1R4(config-msdp)#peer 33.1.1.7

CER7(config)#ip msdp peer 33.56.4.4 connect-source BDI33
```

You should now see a reply from CEL7.

```
RP/0/0/CPU0:SP1R2#ping 233.33.56.2 sou 33.56.2.2 c 40
[..]
Reply to request 2 from 33.56.1.1, 9 ms
Reply to request 2 from 33.56.3.3, 9 ms
Reply to request 2 from 22.1.1.1, 29 ms
Reply to request 3 from 33.56.1.1, 1 ms
Reply to request 3 from 33.56.3.3, 1 ms
Reply to request 3 from 22.1.1.1, 29 ms
```

A packet capture from the interface Gig2 on CER7 is depicted as follows. The packet no. 23 is going through a PW and no. 24 is going through the tunnel2.



In the counters you should have the increasing number of packets received with no increasing RPF fails or drops.

```
CEL7#sh ip mroute vrf csc-cust 233.33.56.2 count
[..]
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group: 233.33.56.2, Source count: 1, Packets forwarded: 0, Packets received: 33
  RP-tree: Forwarding: 0/0/0/0, Other: 3/0/3
  Source: 33.56.2.2/32, Forwarding: 0/0/0/0, Other: 30/0/30

CER7#sh ip mroute 233.33.56.2
[..]
(*, 233.33.56.2), 04:40:14/00:02:40, RP 33.56.10.10, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    Tunnel2, Forward/Sparse, 04:40:14/00:02:40

(33.56.2.2, 233.33.56.2), 00:02:56/00:00:03, flags: MT
  Incoming interface: BDI33, RPF nbr 33.1.1.1
  Outgoing interface list:
    Tunnel2, Forward/Sparse, 00:02:56/00:02:40
```

Verify an MSDP session. There is one active source (SA) learnt from the peer.

```
CER7#sh ip msdp peer
MSDP Peer 33.56.4.4 (?), AS ?
  Connection status:
    State: Up, Resets: 0, Connection source: BDI33 (33.1.1.7)
    Uptime(Downtime): 00:03:41, Messages sent/received: 3/10
    Output messages discarded: 0
    Connection and counters cleared 00:03:41 ago
  SA Filtering:
    Input (S,G) filter: none, route-map: none
    Input RP filter: none, route-map: none
    Output (S,G) filter: none, route-map: none
    Output RP filter: none, route-map: none
  SA-Requests:
    Input filter: none
  Peer ttl threshold: 0
  SAs learned from this peer: 1
  Number of connection transitions to Established state: 1
    Input queue size: 0, Output queue size: 0
  MD5 signature protection on MSDP TCP connection: not enabled
  Message counters:
    RPF Failure count: 0
    SA Messages in/out: 5/0
    SA Requests in: 0
    SA Responses out: 0
    Data Packets in/out: 0/0

RP/0/0/CPU0:SP1R4#sh msdp peer
MSDP Peer 33.1.1.7 (?), AS 0
Description:
  Connection status:
    State: Established, Resets: 2, Connection Source: 33.56.4.4
    Uptime(Downtime): 00:03:56, SA messages received: 0
    TLV messages sent/received: 10/3
  Output messages discarded: 0
    Connection and counters cleared 00:03:56 ago
  SA Filtering:
    Input (S,G) filter: none
    Input RP filter: none
    Output (S,G) filter: none
    Output RP filter: none
  SA-Requests:
    Input filter: none
    Sending SA-Requests to peer: disabled
  Password: None
  Peer ttl threshold: 2
  Input queue size: 0, Output queue size: 0
  KeepAlive timer period: 30
  Peer Timeout timer period: 75
  NSR:
    State: Unknown, Oper-Downs: 0
    NSR-Uptime(NSR-Downtime): 2d01h

CER7#sh ip msdp sa-cache
MSDP Source-Active Cache - 1 entries
(33.56.2.2, 233.33.56.2), RP 33.56.10.10, BGP/AS 0, 00:05:41/00:05:22, Peer 33.56.4.4

RP/0/0/CPU0:SP1R4#sh msdp sa-cache
MSDP Flags:
E - set MRIB E flag , L - domain local source is active,
```

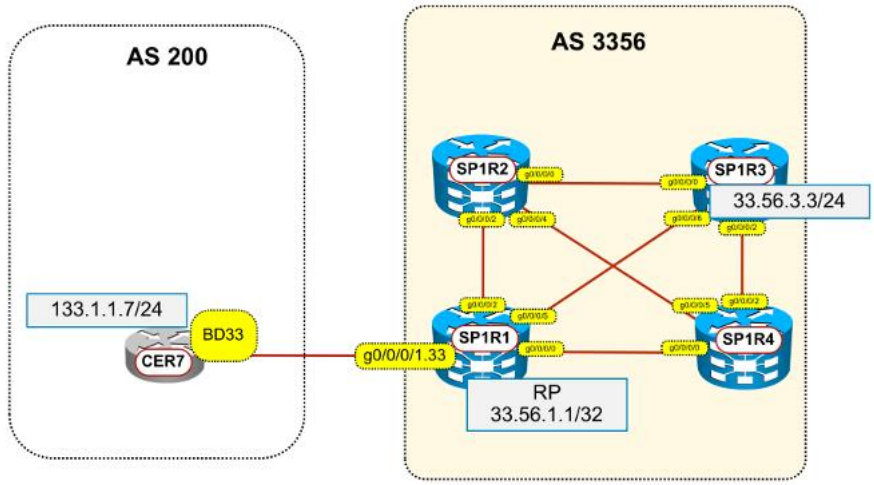

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
EA - externally active source, PI - PIM is interested in the group,
DE - SAs have been denied.  Timers age/expiration,
Cache Entry:
(33.56.2.2, 233.33.56.2), RP 33.56.10.10, MBGP/AS 0, 00:10:32/local
  Learned from peer local, RPF peer local
  SAs recvd 0, Encapsulated data received: 200
    grp flags: PI,  src flags: L
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3: PIM ASM with Static RP and static IGMP Join

Let's assume that CER7 is in one administrative domain and SP1 routers in the other domain. Configure the network so no PIM Join messages are necessary for the group 233.1.1.1 between two domains. At the same time SP1R3 should receive the multicast traffic from 133.1.1.7 without advertising the prefixes to SP1 or keeping the (S,G) entries in the multicast table on SP1R3.



Add a new loopback on CER7 which will be the mcast source. You will use the NAT trick to hide an original source IP address.

```
CER7(config)#int lo133
CER7(config-if)#ip add 133.1.1.7 255.255.255.255
CER7(config)#ip nat inside source static 133.1.1.7 33.1.1.17
CER7(config)#int bdi33
CER7(config-if)#ip nat outside
```

Add the static-group command. With this method you can push the multicast stream without receiving a join message from a neighbor.

```
CER7(config)#interface BDI33
CER7(config-if)#no ip igmp join-group 233.1.1.1
CER7(config-if)#ip igmp static-group 233.1.1.1

CER7#sh ip mroute 233.1.1.1
[..]
(*, 233.1.1.1), 00:00:22/00:02:37, RP 33.56.10.10, flags: SJC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    BDI33, Forward/Sparse, 00:00:22/00:02:37
```

Configure R1 as the RP but just for the group 233.1.1.1. Add the following config on both R1 and R3. The SPT threshold set to infinity means that for the group-list there is no switchover to SPT. So no (S,G) entries are expected on R3 and no need to validate the source against RPF. The RPF check is performed on the RP address.

```
!
ipv4 access-list mcast-233.1.1.1
 10 permit ipv4 host 233.1.1.1 any
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
!
router pim
 address-family ipv4
  rp-address 33.56.1.1 mcast-233.1.1.1
  spt-threshold infinity group-list mcast-233.1.1.1
!

RP/0/0/CPU0:SP1R1#sh mrib route 233.1.1.1
[..]
(*,233.1.1.1) RPF nbr: 33.56.1.1 Flags: L C RPF P
  Up: 2d04h
  Incoming Interface List
    Decapstunnel0 Flags: A DI, Up: 09:01:57
  Outgoing Interface List
    GigabitEthernet0/0/0/5 Flags: F NS, Up: 02:43:02

CER7#ping 233.1.1.1 sou lo133 re 1000
Sending 1000, 100-byte ICMP Echos to 233.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 133.1.1.17
.
Reply to request 1 from 33.56.3.3, 24 ms
Reply to request 2 from 33.56.3.3, 26 ms
Reply to request 3 from 33.56.3.3, 19 ms
Reply to request 4 from 33.56.3.3, 22 ms
```

There is (S,G) entry on R1 with the correct OIL.

```
RP/0/0/CPU0:SP1R1#sh mrib route 233.1.1.1
[..]
(*,233.1.1.1) RPF nbr: 33.56.1.1 Flags: L C RPF P
  Up: 2d04h
  Incoming Interface List
    Decapstunnel0 Flags: A NS DI, Up: 09:06:23
  Outgoing Interface List
    GigabitEthernet0/0/0/5 Flags: F NS, Up: 00:01:29

(33.1.1.17,233.1.1.1) RPF nbr: 33.1.1.17 Flags: RPF
  Up: 00:01:45
  Incoming Interface List
    GigabitEthernet0/0/0/1.33 Flags: A, Up: 00:01:29
  Outgoing Interface List
    GigabitEthernet0/0/0/5 Flags: F NS, Up: 00:01:29
```

There is no (S,G) entry on R3. The counter of forwarded packets is incrementing which is as expected.

```
RP/0/0/CPU0:SP1R3#sh mfib route 233.1.1.1
[..]
(*,233.1.1.1),  Flags:  C
  Up: 09:13:45
  Last Used: 00:00:01
  SW Forwarding Counts: 1231/1231/123100
  SW Replication Counts: 1231/0/0
  SW Failure Counts: 0/0/0/0/0
  Loopback0 Flags:  IC NS EG, Up:09:12:13
  GigabitEthernet0/0/0/6 Flags:  A, Up:09:12:13
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4: PIM ASM with Auto RP

Configure Auto-RP for 233.33.56.1 in SP1. Elect R3 as an RP candidate and mapping agent. Configure R3 as follows.

```
!
ipv4 access-list mcast-233.33.56.1
 10 permit ipv4 host 233.33.56.1 any
!
router pim
 address-family ipv4
  auto-rp mapping-agent Loopback0 scope 5 interval 60
  auto-rp candidate-rp Loopback0 scope 5 group-list mcast-233.33.56.1 interval 60
 !
!
```

The Auto RP election is preferred over the static configuration. If you want to have a static configuration preferred use the keyword 'override' along with the rp-address command.

```
RP/0/0/CPU0:SP1R2#sh pim rp mapping
PIM Group-to-RP Mappings
Group(s) 233.33.56.1/32
  RP 33.56.3.3 (?), v2
    Info source: 33.56.3.3 (?), elected via autorp
      Uptime: 00:01:24, expires: 00:02:41
Group(s) 224.0.0.0/4
  RP 33.56.10.10 (?), v2
    Info source: 0.0.0.0 (?), elected via config
      Uptime: 15:39:20, expires: never
```

Join R1 and R4 to the group.

```
!
router igmp
 interface Loopback0
  join-group 233.33.56.1
 !
```

The status prior sending a ping. The incoming interface points at RP because the very first packets R4 expects to receive from RP.

```
RP/0/0/CPU0:SP1R4#sh mrib route 233.33.56.1
[..]
(*,233.33.56.1) RPF nbr: 33.56.34.3 Flags: C RPF P
  Up: 00:04:12
  Incoming Interface List
    GigabitEthernet0/0/0/2 Flags: A NS, Up: 00:00:39
  Outgoing Interface List
    Loopback0 Flags: F IC NS II LI, Up: 00:00:39
```

After a ping there is a switchover to SPT which is visible in the (S,G) entry.

```
RP/0/0/CPU0:SP1R2#ping 233.33.56.1 sou 33.56.2.2 c 4
Type escape sequence to abort.
Sending 4, 100-byte ICMP Echos to 233.33.56.1, timeout is 2 seconds:
..
Reply to request 2 from 33.56.1.1, 9 ms
Reply to request 2 from 33.56.4.4, 9 ms
Reply to request 3 from 33.56.4.4, 19 ms
Reply to request 3 from 33.56.4.4, 19 ms
Reply to request 3 from 33.56.1.1, 19 ms
Reply to request 3 from 33.56.1.1, 19 ms
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:SP1R4#sh mrib route 233.33.56.1
[..]
(*,233.33.56.1) RPF nbr: 33.56.34.3 Flags: C RPF P
  Up: 00:06:18
  Incoming Interface List
    GigabitEthernet0/0/0/2 Flags: A NS, Up: 00:02:45
  Outgoing Interface List
    Loopback0 Flags: F IC NS II LI, Up: 00:02:45

(33.56.2.2,233.33.56.1) RPF nbr: 33.56.24.2 Flags: RPF
  Up: 00:00:13
  Incoming Interface List
    GigabitEthernet0/0/0/5 Flags: A, Up: 00:00:13
  Outgoing Interface List
    Loopback0 Flags: F IC NS, Up: 00:00:13

RP/0/0/CPU0:SP1R1#sh mrib route 233.33.56.1
[..]
(*,233.33.56.1) RPF nbr: 33.56.13.3 Flags: C RPF P
  Up: 00:07:23
  Incoming Interface List
    GigabitEthernet0/0/0/5 Flags: A NS, Up: 00:04:00
  Outgoing Interface List
    Loopback0 Flags: F IC NS II LI, Up: 00:04:00

(33.56.2.2,233.33.56.1) RPF nbr: 33.56.12.2 Flags: RPF
  Up: 00:01:18
  Incoming Interface List
    GigabitEthernet0/0/0/2 Flags: A, Up: 00:01:18
  Outgoing Interface List
    Loopback0 Flags: F IC NS, Up: 00:01:18
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

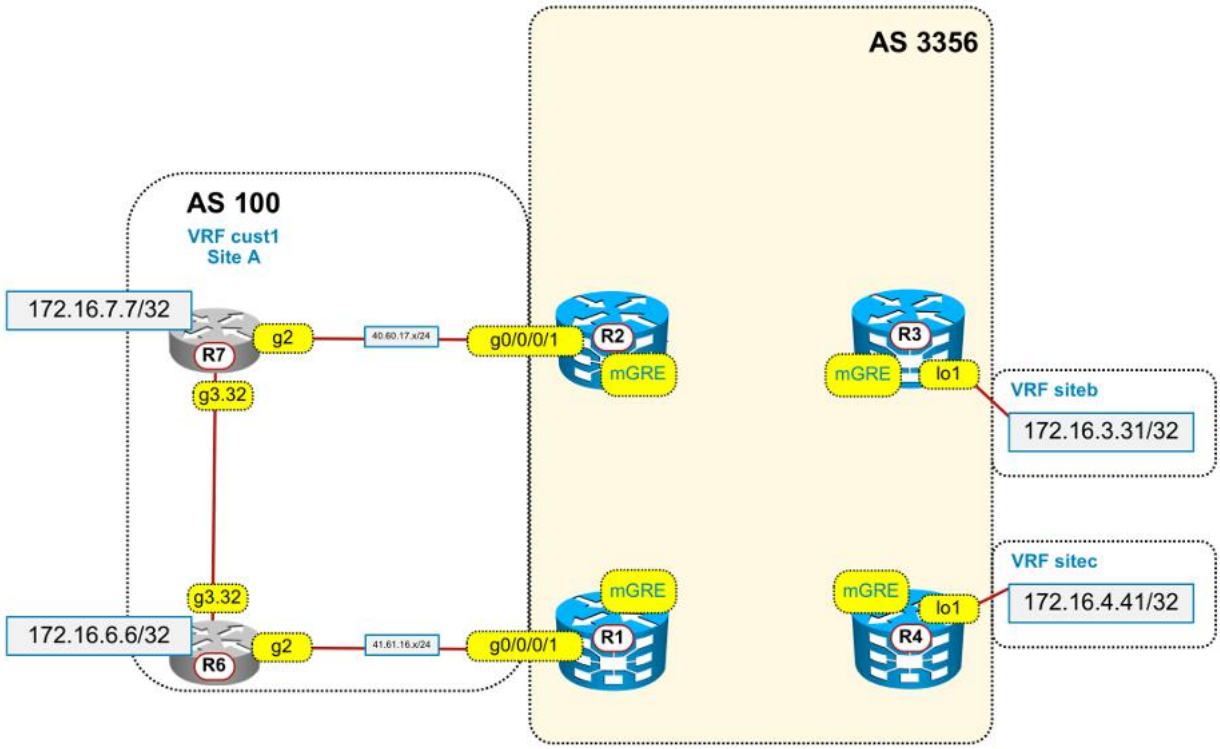
Task 2: mVPN Multicast

Configure devices to allows the multicast traffic between VRFs over the mVPN network.

Task 2.1: Profile 0 - Default MDT, GRE, PIM C-mcast Signaling

If you did the task 1.2 and 1.4 the core network in AS3356 is already prepared for sending the multicast traffic. If not, you need to prepare AS3356 for sending two mcast groups: 233.33.56.1 and 233.33.56.2.

Configure the overlay multicast so the source of the mcast stream is 172.16.3.31 in VRF siteb. The 3 receivers are 172.16.7.7 in VRF cust1, 192.168.1.11 in VRF cust1 and 172.16.4.41 in VRF sitec. In the overlay use Auto-RP PIM SM with R6 as an RP.



Prepare the VRF and MDT configuration in AS3356.

Configure R1 and R2 as follows.

R1, R2:

```
!
route-policy pim1
  set core-tree pim-default
end-policy
!
multicast-routing
 address-family ipv4
  mdt source Loopback0
 !
 vrf cust1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  address-family ipv4
   mdt source Loopback0
   rate-per-route
   interface all enable
   accounting per-prefix
   mdt default ipv4 233.33.56.1
   mdt data 233.33.56.2/32
  !
 !
!
multicast-routing
!
router pim
 vrf cust1
  address-family ipv4
   rpf topology route-policy pim1
   interface GigabitEthernet0/0/0/1
    enable
   !
  !
 !
!
```

Configure R3 as follows.

R3:

```
!
route-policy pim1
  set core-tree pim-default
end-policy
!
multicast-routing
 address-family ipv4
  mdt source Loopback0
 !
 vrf siteb
  address-family ipv4
   mdt source Loopback0
   rate-per-route
   interface all enable
   accounting per-prefix
   mdt default ipv4 233.33.56.1
   mdt data 233.33.56.2/32
  !
 !
!
multicast-routing
!
router pim
 vrf siteb
  address-family ipv4
   rpf topology route-policy pim1
   interface lo1
    enable
   !
  !
 !
!
```

Configure R4.

R4:

```
!
route-policy pim1
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  set core-tree pim-default
end-policy
!
multicast-routing
 address-family ipv4
  mdt source Loopback0
 !
 vrf sitec
  address-family ipv4
   mdt source Loopback0
   rate-per-route
   interface all enable
   accounting per-prefix
   mdt default ipv4 233.33.56.1
   mdt data 233.33.56.2/32
  !
 !
!
multicast-routing
!
router pim
 vrf sitec
  address-family ipv4
   rpf topology route-policy pim1
   interface lo1
    enable
   !
  !
 !
!
```

The underlay Multicast Distribution Tree (MDT) based on mGRE is ready. Compare the following output with other routers in AS3356.

```
RP/0/0/CPU0:SP1R2#sh mrib route 233.33.56.1
[..]
(*,233.33.56.1) RPF nbr: 33.56.23.3 Flags: C RPF MD MH CD P
  MVPN TID: 0xe0000013
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x190
  Up: 00:29:49
  Incoming Interface List
    GigabitEthernet0/0/0/0 Flags: A NS, Up: 00:09:26
  Outgoing Interface List
    Loopback0 Flags: F NS, Up: 00:09:26

(33.56.1.1,233.33.56.1) RPF nbr: 33.56.12.1 Flags: RPF MD MH CD
  MVPN TID: 0xe0000013
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x190
  Up: 00:09:17
  Incoming Interface List
    GigabitEthernet0/0/0/2 Flags: A, Up: 00:09:17
  Outgoing Interface List
    Loopback0 Flags: F NS, Up: 00:09:17

(33.56.2.2,233.33.56.1) RPF nbr: 33.56.2.2 Flags: RPF ME MH
  MVPN TID: 0xe0000013
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x190
  Up: 00:09:26
  Incoming Interface List
    Loopback0 Flags: F A, Up: 00:09:26
  Outgoing Interface List
    Loopback0 Flags: F A, Up: 00:09:26
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    GigabitEthernet0/0/0/0 Flags: F NS, Up: 00:08:12
    GigabitEthernet0/0/0/2 Flags: F NS, Up: 00:09:19
    GigabitEthernet0/0/0/4 Flags: F NS, Up: 00:09:19

(33.56.3.3,233.33.56.1) RPF nbr: 33.56.23.3 Flags: RPF MD MH CD
  MVPN TID: 0xe0000013
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x190
  Up: 00:08:06
  Incoming Interface List
    GigabitEthernet0/0/0/0 Flags: A, Up: 00:08:06
  Outgoing Interface List
    Loopback0 Flags: F NS, Up: 00:08:06

(33.56.4.4,233.33.56.1) RPF nbr: 33.56.24.4 Flags: RPF MD MH CD
  MVPN TID: 0xe0000013
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x190
  Up: 00:00:10
  Incoming Interface List
    GigabitEthernet0/0/0/4 Flags: A, Up: 00:00:10
  Outgoing Interface List
    Loopback0 Flags: F NS, Up: 00:00:10
```

Now, let's prepare the multicast network in the overlay.

Configure PIM and RP.

```
RP/0/0/CPU0:SP1R4(config)#router igmp
RP/0/0/CPU0:SP1R4(config-igmp)#vrf sitec
RP/0/0/CPU0:SP1R4(config-igmp-sitec)#int lo1
RP/0/0/CPU0:SP1R4(config-igmp-sitec-if)#join-group 237.1.1.1

RP/0/0/CPU0:SP1R1(config)#router igmp
RP/0/0/CPU0:SP1R1(config-igmp)#vrf sitec
RP/0/0/CPU0:SP1R1(config-igmp-sitec)#int lo3
RP/0/0/CPU0:SP1R1(config-igmp-sitec-if)#join-group 237.1.1.1

CEL7(config)#ip multicast-routing distributed
CEL7(config)#ip multicast-routing vrf csc-cust distributed
CEL7(config)#int g2
CEL7(config-if)#ip pim sparse-m
CEL7(config-if)#int g3.32
CEL7(config-subif)#ip pim sparse-m
CEL7(config-subif)#int lo0
CEL7(config-if)#ip pim sparse-m
CEL7(config-if)#ip igmp join-group 237.1.1.1
CEL7(config)#ip pim autorp listener

CEL6(config)#ip multicast-routing distributed
CEL6(config)#int g3.32
CEL6(config-subif)#ip pim sparse-m
CEL6(config-subif)#int lo0
CEL6(config-if)#ip pim sparse-m
CEL6(config)#ip pim send-rp-announce Loopback0 scope 10
CEL6(config)#ip pim send-rp-discovery Loopback0 scope 10
```

Verify routing towards the RP.

```
RP/0/0/CPU0:SP1R3#sh route vrf siteb 172.16.6.6
Routing entry for 172.16.6.6/32
  Known via "bgp 3356", distance 200, metric 3, type internal
  Installed May 28 01:20:58.604 for 00:22:30
  Routing Descriptor Blocks
    33.56.2.2, from 33.56.5.5
      Nexthop in Vrf: "default", Table: "default", IPv4 Unicast, Table Id: 0xe0000000
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Route metric is 3
  No advertising protos.

RP/0/0/CPU0:SP1R4#sh route vrf sitec 172.16.6.6
Routing entry for 172.16.6.6/32
  Known via "bgp 3356", distance 200, metric 3, type internal
  Installed May 28 01:20:58.485 for 00:22:10
  Routing Descriptor Blocks
    33.56.2.2, from 33.56.5.5
      Nexthop in Vrf: "default", Table: "default", IPv4 Unicast, Table Id: 0xe0000000
      Route metric is 3
  No advertising protos.

CEL7#sh ip cef 172.16.6.6
172.16.6.6/32
  nexthop 10.1.67.6 GigabitEthernet3.32
```

Prepare the inter-AS connection on R3. The following part of the configuration is needed if there is no RP learnt within the VRF siteb and sitec. The loopback1 alone may not trigger the VRF to join the MDT default.

R3:

```
!
interface GigabitEthernet0/0/0/1.35
 vrf siteb
 ipv4 address 22.22.35.1 255.255.255.0
 encapsulation dot1q 35
!
router pim
 vrf siteb
  address-family ipv4
   interface GigabitEthernet0/0/0/1.35
    enable
   !
  !
 !
!
```

R4:

```
!
interface GigabitEthernet0/0/0/1.35
 vrf sitec
 ipv4 address 11.11.35.1 255.255.255.0
 encapsulation dot1q 35
!
router pim
 vrf sitec
  address-family ipv4
   interface GigabitEthernet0/0/0/1.35
    enable
   !
  !
 !
!
```

Verify that you can see the RP's address 172.16.6.6 in the overlay.

```
RP/0/0/CPU0:SP1R1#sh pim vrf cust1 rp mapping
PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 172.16.6.6 (?), v2
    Info source: 172.16.6.6 (?), elected via autorp
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Uptime: 05:00:34, expires: 00:02:19

RP/0/0/CPU0:SP1R3#sh pim vrf siteb rp mapping
PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 172.16.6.6 (?), v2
    Info source: 172.16.6.6 (?), elected via autorp
      Uptime: 02:25:09, expires: 00:02:56

RP/0/0/CPU0:SP1R4#sh pim vrf sitec rp mapping
PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 172.16.6.6 (?), v2
    Info source: 172.16.6.6 (?), elected via autorp
      Uptime: 00:03:15, expires: 00:02:42
```

Check the multicast table. Entries for group 224.0.1.39 and 224.0.1.40 are related to Auto-RP received via a default MDT.

```
RP/0/0/CPU0:SP1R3#sh mrib vrf siteb route
[..]
(*,224.0.0.0/4) RPF nbr: 33.56.2.2 Flags: C RPF P
  Up: 02:26:11

(*,224.0.0.0/24) Flags: D P
  Up: 03:25:41

(*,224.0.1.39) Flags: S P
  Up: 03:25:41

(172.16.6.6,224.0.1.39) RPF nbr: 33.56.2.2 Flags: RPF
  Up: 02:36:53
  Incoming Interface List
    mdtsiteb Flags: F A MI, Up: 02:36:53
  Outgoing Interface List
    mdtsiteb Flags: F A MI, Up: 02:36:53
    Loopback1 Flags: F, Up: 02:36:53
    GigabitEthernet0/0/0/1.35 Flags: F, Up: 02:26:12

(*,224.0.1.40) Flags: S P
  Up: 03:25:41
  Outgoing Interface List
    GigabitEthernet0/0/0/1.35 Flags: II LI, Up: 02:26:12

(172.16.6.6,224.0.1.40) RPF nbr: 33.56.2.2 Flags: RPF
  Up: 02:37:07
  Incoming Interface List
    mdtsiteb Flags: F A MI, Up: 02:37:07
  Outgoing Interface List
    mdtsiteb Flags: F A MI, Up: 02:37:07
    Loopback1 Flags: F, Up: 02:37:07
    GigabitEthernet0/0/0/1.35 Flags: F IC, Up: 02:26:12

(*,232.0.0.0/8) Flags: D P
  Up: 03:25:41
```

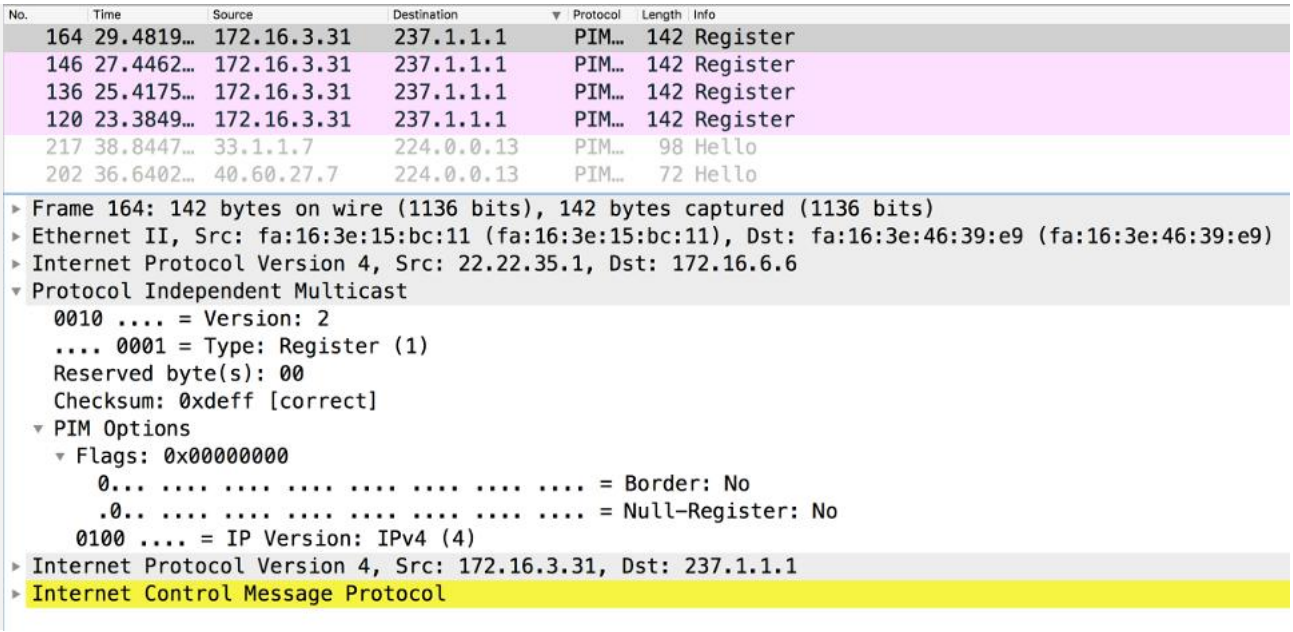
Test the multicast transmission.

```
RP/0/0/CPU0:SP1R3#ping vrf siteb 237.1.1.1 source lo1 re 5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 237.1.1.1, timeout is 2 seconds:
.
Reply to request 1 from 192.168.1.11, 9 ms
Reply to request 1 from 172.16.7.7, 9 ms
Reply to request 2 from 192.168.1.11, 9 ms
Reply to request 3 from 192.168.1.11, 1 ms
Reply to request 3 from 172.16.7.7, 9 ms
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Reply to request 4 from 192.168.1.11, 9 ms
Reply to request 4 from 172.16.7.7, 9 ms
```

A PIM debug on the RP shows a registration of a new (S,G). This message is the unicast packet encapsulating the multicast (see a packet capture depicted below). Then PIM Join is sent towards the source 172.16.3.31 hop by hop. The source sends the native multicast stream which stops the encapsulated traffic and triggers a switchover to the SPT and pruning the RPT (Rendezvous Point Tree).



A debug output from R6 (RP).

```
*May 19 23:02:28.128: PIM(0): Received v2 Register on GigabitEthernet3.32 from
172.16.3.31
*May 19 23:02:28.129:      for 172.16.3.31, group 237.1.1.1
*May 19 23:02:28.130: PIM(0): Adding register decap tunnel (Tunnel1) as accepting
interface of (172.16.3.31, 237.1.1.1).
*May 19 23:02:28.130: PIM(0): Insert (172.16.3.31,237.1.1.1) join in nbr 10.1.67.7's
queue
*May 19 23:02:28.131: PIM(0): Building Join/Prune packet for nbr 10.1.67.7
*May 19 23:02:
CEL6#28.131: PIM(0):  Adding v2 (172.16.3.31/32, 237.1.1.1), S-bit Join
*May 19 23:02:28.131: PIM(0): Send v2 join/prune to 10.1.67.7 (GigabitEthernet3.32)
*May 19 23:02:28.150: PIM(0): Received v2 Assert on GigabitEthernet3.32 from 10.1.67.7
*May 19 23:02:28.150: PIM(0): Assert metric to source 172.16.3.31 is [110/1]
*May 19 23:02:28.150: PIM(0): Cached metric is RP-bit [Inf/4294967295]
*May 19 23:02:28.233: PIM(0): Received v2 Join/Prune on GigabitEthernet3.32 from
10.1.67.7, to us
*May 19 23:02:28.233:
CEL6# PIM(0): Join-list: (*, 237.1.1.1), RPT-bit set, WC-bit set, S-bit set
*May 19 23:02:28.234: PIM(0): Update GigabitEthernet3.32/10.1.67.7 to (*, 237.1.1.1),
Forward state, by PIM *G Join
*May 19 23:02:28.234: PIM(0): Prune-list: (172.16.3.31/32, 237.1.1.1) RPT-bit set
*May 19 23:02:30.019: PIM(0): check pim_rp_announce 1
*May 19 23:02:30.020: PIM(0): send rp announce
*May 19 23:02:30.469: PIM(0): Received v2 Register on GigabitEthernet3.32 from
172.16.3.31
*May 19 23:02:30.470:      for 172.16.3.31,
CEL6#group 237.1.1.1
*May 19 23:02:30.485: PIM(0): Removing register decap tunnel (Tunnel1) as accepting
interface of (172.16.3.31, 237.1.1.1).
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
*May 19 23:02:30.486: PIM(0): Installing GigabitEthernet3.32 as accepting interface for
(172.16.3.31, 237.1.1.1).
*May 19 23:02:30.528: PIM(0): Insert (172.16.3.31,237.1.1.1) join in nbr 10.1.67.7's
queue
*May 19 23:02:30.528: PIM(0): Building Periodic (*,G) Join / (S,G,RP-bit) Prune message
for 237.1.1.1
*May 19 23:02:30.528: PIM(0): Building Join/Prune packet for
CEL6# nbr 10.1.67.7
*May 19 23:02:30.528: PIM(0):  Adding v2 (172.16.3.31/32, 237.1.1.1), S-bit Join
*May 19 23:02:30.528: PIM(0): Send v2 join/prune to 10.1.67.7 (GigabitEthernet3.32)
*May 19 23:02:32.758: PIM(0): Received v2 Register on GigabitEthernet3.32 from
172.16.3.31
*May 19 23:02:32.758:      for 172.16.3.31, group 237.1.1.1
*May 19 23:02:32.758: PIM(0): Send v2 Register-Stop to 172.16.3.31 for 172.16.3.31,
group 237.1.1.1
*May 19 23:02:32.826: PIM(0): Insert (172.16.3.31,237.1.1.1) prune in nbr
CEL6#10.1.67.7's queue
*May 19 23:02:32.830: PIM(0): Building Join/Prune packet for nbr 10.1.67.7
*May 19 23:02:32.830: PIM(0):  Adding v2 (172.16.3.31/32, 237.1.1.1), S-bit Prune
*May 19 23:02:32.830: PIM(0): Send v2 join/prune to 10.1.67.7 (GigabitEthernet3.32)
*May 19 23:02:32.837: PIM(0): Received v2 Assert on GigabitEthernet3.32 from 10.1.67.7
*May 19 23:02:32.838: PIM(0): Assert metric to RP 172.16.3.31 is [Inf/1]
*May 19 23:02:32.838: PIM(0): Cached metric is [110/1]
```

Check the multicast tables and (S,G) entries which appeared on routers. The multicast group number identifies the VRF so the router knows where to send incoming packets.

```
RP/0/0/CPU0:SP1R3#sh mrib vrf siteb route 237.1.1.1
[..]
(172.16.3.31,237.1.1.1) RPF nbr: 172.16.3.31 Flags: RPF
  Up: 00:01:29
  Incoming Interface List
    Loopback1 Flags: A, Up: 00:01:29
  Outgoing Interface List
    mdtsiteb Flags: F NS MI, Up: 00:01:27

RP/0/0/CPU0:SP1R1#sh mrib vrf cust1 route 237.1.1.1
[..]
(*,237.1.1.1) RPF nbr: 33.56.2.2 Flags: C RPF
  Up: 02:28:47
  Incoming Interface List
    mdtcust1 Flags: A NS MI, Up: 02:28:47
  Outgoing Interface List
    Loopback3 Flags: F IC NS II LI, Up: 02:28:47

(172.16.3.31,237.1.1.1) RPF nbr: 33.56.3.3 Flags: RPF
  Up: 00:02:53
  Incoming Interface List
    mdtcust1 Flags: A MI, Up: 00:02:53
  Outgoing Interface List
    Loopback3 Flags: F IC NS, Up: 00:02:53

RP/0/0/CPU0:SP1R4#sh mrib vrf sitec route 237.1.1.1
[..]
(*,237.1.1.1) RPF nbr: 33.56.2.2 Flags: C RPF
  Up: 03:19:56
  Incoming Interface List
    mdtsitec Flags: A NS MI, Up: 00:12:51
  Outgoing Interface List
    Loopback1 Flags: F IC NS II LI, Up: 03:19:56

(172.16.3.31,237.1.1.1) RPF nbr: 33.56.2.2 Flags: RPF
  Up: 00:03:40
  Incoming Interface List
    mdtsitec Flags: A MI, Up: 00:03:40
  Outgoing Interface List
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    Loopback1 Flags: F IC NS, Up: 00:03:40

CEL6#sh ip mroute 237.1.1.1
[..]
(*, 237.1.1.1), 03:18:22/00:03:12, RP 172.16.6.6, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet3.32, Forward/Sparse, 03:18:22/00:03:12

(172.16.3.31, 237.1.1.1), 00:00:17/00:02:48, flags: PT
  Incoming interface: GigabitEthernet3.32, RPF nbr 10.1.67.7
  Outgoing interface list: Null

CEL7#sh ip mroute 237.1.1.1
[..]
(*, 237.1.1.1), 06:54:01/00:02:56, RP 172.16.6.6, flags: SJCL
  Incoming interface: GigabitEthernet3.32, RPF nbr 10.1.67.6
  Outgoing interface list:
    GigabitEthernet2, Forward/Sparse, 02:27:35/00:02:56
    Loopback0, Forward/Sparse, 06:53:11/00:02:30

(172.16.3.31, 237.1.1.1), 00:01:40/00:01:19, flags: LT
  Incoming interface: GigabitEthernet2, RPF nbr 40.60.27.2
  Outgoing interface list:
    Loopback0, Forward/Sparse, 00:01:40/00:02:30
```

As you could notice there is no response from R4. This is caused by a lack of the multicast source reachability in VRF sitec. You can fix it by playing with the RT settings.

```
RP/0/0/CPU0:SP1R4(config)#vrf sitec
RP/0/0/CPU0:SP1R4(config-vrf)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R4(config-vrf-af)#import route-target
RP/0/0/CPU0:SP1R4(config-vrf-import-rt)#3:3
```

Now, there are responses from 3 receivers.

```
RP/0/0/CPU0:SP1R3#ping vrf siteb 237.1.1.1 source lo1 re 5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 237.1.1.1, timeout is 2 seconds:

Reply to request 0 from 192.168.1.11, 1 ms
Reply to request 0 from 172.16.4.41, 1 ms
Reply to request 1 from 172.16.4.41, 9 ms
Reply to request 1 from 192.168.1.11, 9 ms
Reply to request 1 from 172.16.7.7, 9 ms
Reply to request 2 from 192.168.1.11, 9 ms
Reply to request 2 from 172.16.4.41, 9 ms
Reply to request 2 from 172.16.7.7, 39 ms
Reply to request 3 from 172.16.4.41, 1 ms
Reply to request 3 from 192.168.1.11, 1 ms
Reply to request 3 from 172.16.7.7, 9 ms
Reply to request 4 from 172.16.4.41, 9 ms
Reply to request 4 from 192.168.1.11, 9 ms
Reply to request 4 from 172.16.7.7, 9 ms
```

In real life the multicast source doesn't need to receive responses but still a last hop router needs the source IP information to switchover to the SPT and to validate the source against RPF. That's why the correct routing entries or at least route multicast entries are required.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The RPF check against the mcast source.

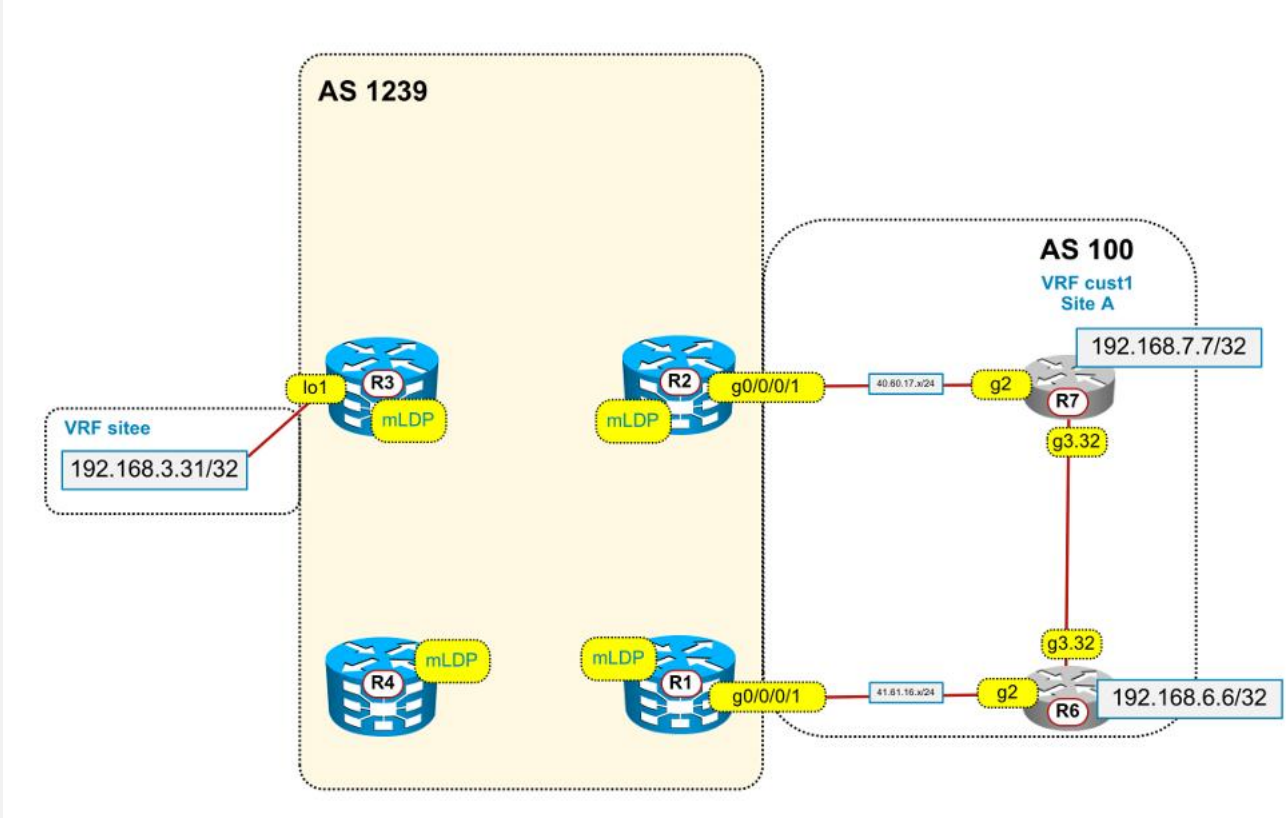
```
RP/0/0/CPU0:SP1R4#sh pim vrf sitec rpf 172.16.3.31

Table: IPv4-Unicast-default
* 172.16.3.31/32 [200/0]
    via mdtsitec with rpf neighbor 33.56.3.3
    Connector: 3:3:33.56.3.3, Nexthop: 33.56.3.3
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.2: Profile 6 - MLDP, In-Band Signaling

In this task you will configure the mVPN profile with mLDP in the core. The customer's signalization coming from PIM to be transmitted via in-band mLDP signalization.





RFC 6826 - In-Band Signaling with mLDP
"When an IP multicast tree (either a source-specific tree or a bidirectional tree) enters the MPLS network, the (S,G) or (*,G) information from the IP multicast control-plane state is carried in the opaque value field of the mLDP FEC message."

Configure mLDP on R1, R2 as the multicast receivers and R3 as the multicast source.

Use the following configuration.

R1, R2:

```
!
ipv4 access-list ssm1
 10 permit ipv4 host 238.1.1.1 any
!
!
route-policy mldp1
  set core-tree mldp-inband
end-policy
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
mpls ldp
 mldp
  logging notifications
  address-family ipv4
  !
 !
!
multicast-routing
 address-family ipv4
  interface Loopback0
   enable
  !
  mdt source Loopback0
 !
 vrf cust1
  address-family ipv4
   mdt source Loopback0
   interface all enable
   mdt mldp in-band-signaling ipv4
  !
 !
!
router pim
 vrf cust1
  address-family ipv4
   rpf topology route-policy mldp1
   interface GigabitEthernet0/0/0/1
    enable
   !
   ssm range ssm1
  !
 !
!
```

R3:

```
!
ipv4 access-list ssm1
 10 permit ipv4 host 238.1.1.1 any
!
!
route-policy mldp1
  set core-tree mldp-inband
end-policy
!
mpls ldp
 mldp
  logging notifications
  address-family ipv4
  !
 !
!
multicast-routing
 address-family ipv4
  interface Loopback0
   enable
  !
  mdt source Loopback0
 !
 vrf sitee
  address-family ipv4
   mdt source Loopback0
   interface all enable
   mdt mldp in-band-signaling ipv4
  !
 !
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
router pim
 vrf sitee
  address-family ipv4
   rpf topology route-policy mldp1
   interface Loopback1
    enable
   !
   ssm range ssm1
  !
 !
!
```

Verify mLDP neighbors. The peer addr list contains IP addresses from the global interfaces.

```
RP/0/0/CPU0:SP2R3#sh mpls mldp neighbors
mLDP neighbor database
 MLDP peer ID      : 12.39.1.1:0, uptime 00:03:42 Up,
  Capabilities     : Typed Wildcard FEC, P2MP, MP2MP
  Target Adj       : No
  Upstream count   : 0
  Branch count     : 0
  Label map timer  : never
  Policy filter in :
  Path count       : 1
  Path(s)          : 12.39.13.1        GigabitEthernet0/0/0/6 LDP
  Adj list         : 12.39.13.1        GigabitEthernet0/0/0/6
  Peer addr list   : 12.39.14.1
                   : 12.39.12.1
                   : 12.39.15.1
                   : 12.39.13.1
                   : 20.40.17.1
                   : 31.1.1.6
                   : 60.1.1.2
                   : 101.1.1.2
                   : 12.39.1.1

 MLDP peer ID      : 12.39.2.2:0, uptime 00:03:42 Up,
  Capabilities     : Typed Wildcard FEC, P2MP, MP2MP
  Target Adj       : No
  Upstream count   : 0
  Branch count     : 0
  Label map timer  : never
  Policy filter in :
  Path count       : 1
  Path(s)          : 12.39.23.2        GigabitEthernet0/0/0/0 LDP
  Adj list         : 12.39.23.2        GigabitEthernet0/0/0/0
  Peer addr list   : 12.39.23.2
                   : 12.39.12.2
                   : 12.39.24.2
                   : 21.41.26.2
                   : 12.39.2.2

 MLDP peer ID      : 12.39.4.4:0, uptime 00:03:42 Up,
  Capabilities     : Typed Wildcard FEC, P2MP, MP2MP
  Target Adj       : No
  Upstream count   : 0
  Branch count     : 0
  Label map timer  : never
  Policy filter in :
  Path count       : 1
  Path(s)          : 12.39.34.4        GigabitEthernet0/0/0/2 LDP
  Adj list         : 12.39.34.4        GigabitEthernet0/0/0/2
  Peer addr list   : 12.39.14.4
                   : 11.11.11.254
                   : 12.39.34.4
                   : 10.0.128.4
                   : 21.41.46.4
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
                   : 12.39.24.4
                   : 12.39.4.4

 MLDP peer ID      : 12.39.5.5:0, uptime 00:03:42 Up,
  Capabilities     : Typed Wildcard FEC, P2MP, MP2MP
  Target Adj       : No
  Upstream count   : 0
  Branch count     : 0
  Label map timer  : never
  Policy filter in :
  Path count       : 1
  Path(s)          : 12.39.35.5        GigabitEthernet0/0/0/3 LDP
  Adj list         : 12.39.35.5        GigabitEthernet0/0/0/3
  Peer addr list   : 12.39.5.5
                   : 12.39.35.5
                   : 12.39.15.5
```

There are no labels assigned yet as no PIM joins from a customer.

```
RP/0/0/CPU0:SP2R3#sh mpls mldp bindings
No entries in the table to display
```

Configure the customer side. Use Source Specific Multicast (SSM).

```
CER7(config)#int lo0
CER7(config-if)#ip pim sparse-m
CER7(config-if)#ip igmp join-group 238.1.1.1
CER7(config-if)#int g2
CER7(config-if)#ip pim sparse-m
```

SSM requires a declaration the mcast range if it different than 232/8. The ssm-map is needed if a receiver does not support version 3 of IGMP.

```
CER7(config)#ip access-list standard mcast-238
CER7(config-std-nacl)#permit host 238.1.1.1
CER7(config-std-nacl)#ip pim ssm range mcast-238
CER7(config)#ip igmp ssm-map enable
Warning: No DNS server is configured. DNS-based SSM mapping
should be disabled if no DNS server is configured.
CER7(config)#ip igmp ssm-map static mcast-238 192.168.3.31
```

Once SSM is configured the PIM join is sent towards the source 192.168.3.31. In the core there is no PIM and mLDP is responsible to deliver the join message to the multicast source.

You can see what labels are allocated. Based on the following example you can see the mcast stream will go out from R3 to R2 with the label 123229.

```
RP/0/0/CPU0:SP2R3#sh mpls mldp bindings
mLDP MPLS Bindings database

LSP-ID: 0x00001 Paths: 2 Flags:
 0x00001 P2MP   12.39.3.3 [vpnv4 200:3 192.168.3.31 238.1.1.1]
   Local Label: 123307
   Remote Label: 123229 NH: 12.39.23.2 Inft: GigabitEthernet0/0/0/0

RP/0/0/CPU0:SP2R3# sh mpls mldp forwarding
mLDP MPLS forwarding database

123307  LSM-ID: 0x00001 HLI: 0x00001 flags: In
   123229, NH: 12.39.23.2, Intf: GigabitEthernet0/0/0/0 Role: H
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Then a P2MP labelled tree can be built with a root at R3.

```
RP/0/0/CPU0:SP2R3#sh mpls mldp root
mLDP root database
 Root node     : 12.39.3.3  (We are the root)
  Metric       : 0
  Distance     : 0
  FEC count    : 1
  Path count   : 1
  Path(s)      : 12.39.3.3        LDP nbr: none
```

Add an another receiver.

```
RP/0/0/CPU0:SP2R1(config)#router igmp
RP/0/0/CPU0:SP2R1(config-igmp)#vrf cust1
RP/0/0/CPU0:SP2R1(config-igmp-cust1)#version 3
RP/0/0/CPU0:SP2R1(config-igmp-cust1)#int g0/0/0/1
RP/0/0/CPU0:SP2R1(config-igmp-cust1-if)#join-group 238.1.1.1 192.168.3.31
```

Now, there is one binding more to R1.

```
RP/0/0/CPU0:SP2R3#sh mpls mldp bindings
mLDP MPLS Bindings database

LSP-ID: 0x00001 Paths: 3 Flags:
 0x00001 P2MP   12.39.3.3 [vpnv4 200:3 192.168.3.31 238.1.1.1]
   Local Label: 123307
   Remote Label: 123229 NH: 12.39.23.2 Inft: GigabitEthernet0/0/0/0
   Remote Label: 123111 NH: 12.39.13.1 Inft: GigabitEthernet0/0/0/6
```

The new labelled path is reflected in the forwarding table.

```
RP/0/0/CPU0:SP2R3#sh mpls for p2mp
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
123307 123111      mLDP/IR: 0x00001  Gi0/0/0/6    12.39.13.1      0
       123229      mLDP/IR: 0x00001  Gi0/0/0/0    12.39.23.2      0
```

Test your connectivity.

```
RP/0/0/CPU0:SP2R3#ping vrf sitee 238.1.1.1 sou lo1 re 5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 238.1.1.1, timeout is 2 seconds:
.
Reply to request 1 from 21.41.16.1, 1 ms
Reply to request 2 from 21.41.16.1, 9 ms
Reply to request 3 from 21.41.16.1, 1 ms
Reply to request 4 from 21.41.16.1, 1 ms
```

As you can see there a response from just one receiver but both receivers get the multicast.

```
CER7#sh ip mroute 238.1.1.1
[..]
(192.168.3.31, 238.1.1.1), 00:07:20/00:02:40, flags: sTI
  Incoming interface: GigabitEthernet2, RPF nbr 20.40.27.2
  Outgoing interface list:
    Loopback0, Forward/Sparse, 00:07:20/00:02:40

CER7#sh ip mroute 238.1.1.1 count
[..]
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Group: 238.1.1.1, Source count: 1, Packets forwarded: 4, Packets received: 4
  Source: 192.168.3.31/32, Forwarding: 4/0/114/0, Other: 4/0/0
```

To receive ICMP replies you can change the configuration as follows.

```
CER7(config)#int lo0
CER7(config-if)#ip igmp join-group 238.1.1.1 sou 192.168.3.31

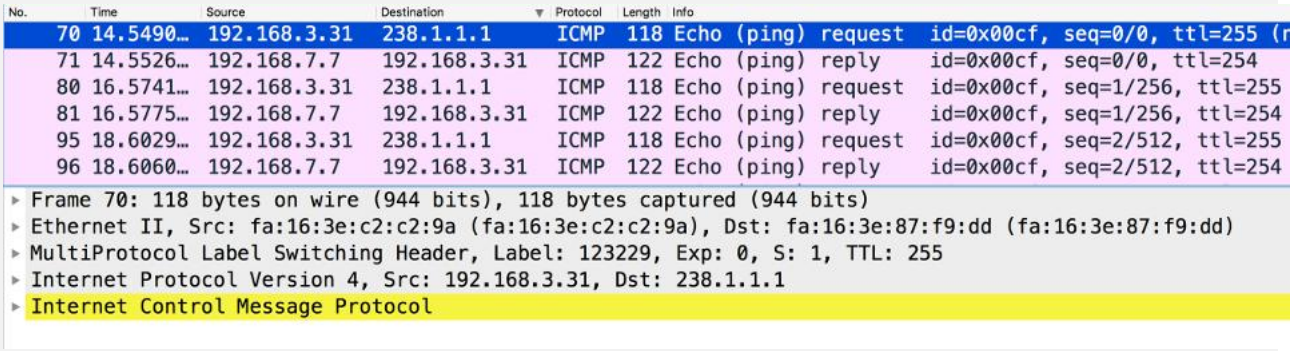
RP/0/0/CPU0:SP2R3#ping vrf sitee 238.1.1.1 sou lo1 re 5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 238.1.1.1, timeout is 2 seconds:

Reply to request 0 from 21.41.16.1, 1 ms
Reply to request 0 from 192.168.7.7, 1 ms
Reply to request 1 from 21.41.16.1, 1 ms
Reply to request 1 from 192.168.7.7, 1 ms
Reply to request 2 from 21.41.16.1, 1 ms
Reply to request 2 from 192.168.7.7, 1 ms
Reply to request 3 from 21.41.16.1, 1 ms
Reply to request 3 from 192.168.7.7, 1 ms
Reply to request 4 from 21.41.16.1, 9 ms
Reply to request 4 from 192.168.7.7, 9 ms
```

Verify outgoing labels imposed to the multicast packets.

```
RP/0/0/CPU0:SP2R3#sh mpls mldp forwarding
mLDP MPLS forwarding database

123307  LSM-ID: 0x00001 HLI: 0x00001 flags: In
   123111, NH: 12.39.13.1, Intf: GigabitEthernet0/0/0/6 Role: H
   123229, NH: 12.39.23.2, Intf: GigabitEthernet0/0/0/0 Role: H
```



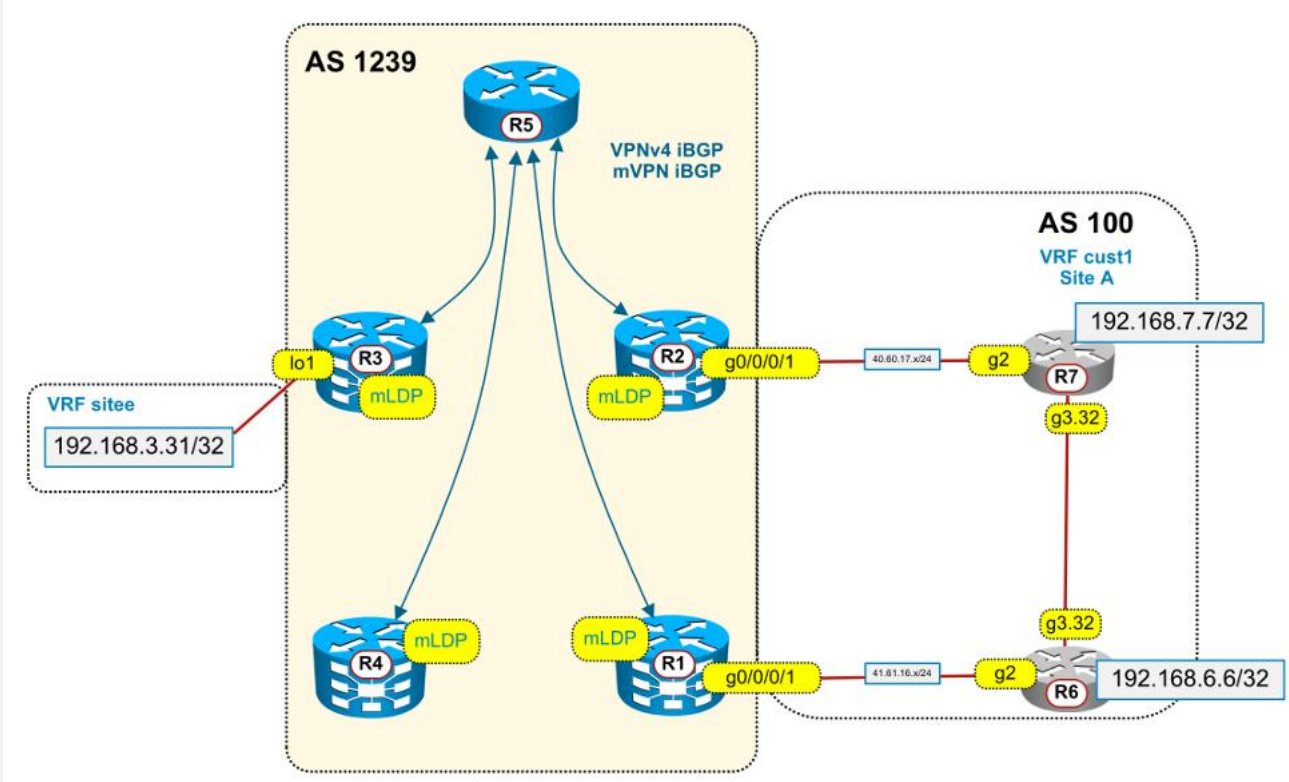
You can check the LSP and the VRF destination name on the edge router.

```
RP/0/0/CPU0:SP2R2#sh mpls mldp lsm-id
mLDP database
LSM-ID: 0x00001  VRF: default  Type: P2MP  Uptime: 00:41:38
  FEC Root           : 12.39.3.3
  Opaque decoded     : [vpnv4 200:3 192.168.3.31 238.1.1.1]
  Upstream neighbor(s) :
    12.39.3.3:0 [Active] Uptime: 00:41:38
      Local Label (D) : 123229
  Downstream  client(s):
    PIM MDT            Uptime: 00:41:38
      Egress intf     : Imdtcust1
      Table ID        : IPv4: 0xe0000012
      RPF ID          : 3
      RD              : 200:3
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.3: Profile 12 - MLDP, P2MP, BGP AD + C-mcast Signaling

In this task you will configure the mVPN profile with mLDP in the core and BGP as a signaling protocol in the overlay network. The customer signalization is provided by PIM SSM on the customer side and within VRFs. Then PIM messages are translated to the BGP messages and transported over the core network.



Prepare your core of the network as follows.

Changes the previous task 2.2 are highlighted.

R1, R2:

```
!
vrf cust1
 vpn id 1:100
!
ipv4 access-list ssm1
 10 permit ipv4 host 238.1.1.1 any
!
!
route-policy mldp1
  set core-tree mldp-default
end-policy
!
mpls ldp
 mldp
  logging notifications
  address-family ipv4
  !
 !
!
multicast-routing
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
 address-family ipv4
  interface Loopback0
   enable
  !
  mdt source Loopback0
 !
 vrf cust1
  address-family ipv4
   mdt source Loopback0
   mdt default mldp p2mp
   interface all enable
   bgp auto-discovery mldp
  !
 !
!
router pim
 vrf cust1
  address-family ipv4
   rpf topology route-policy mldp1
   mdt c-multicast-routing bgp
   interface GigabitEthernet0/0/0/1
    enable
   !
   ssm range ssm1
  !
 !
!
```

R3:

```
!
vrf cust1
 vpn id 1:100
!
ipv4 access-list ssm1
 10 permit ipv4 host 238.1.1.1 any
!
!
route-policy mldp1
  set core-tree mldp-default
end-policy
!
mpls ldp
 mldp
  logging notifications
  address-family ipv4
  !
 !
!
multicast-routing
 address-family ipv4
  interface Loopback0
   enable
  !
  mdt source Loopback0
 !
 vrf sitee
  address-family ipv4
   mdt source Loopback0
   mdt default mldp p2mp
   interface all enable
   bgp auto-discovery mldp
  !
 !
!
router pim
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
 vrf sitee
  address-family ipv4
   rpf topology route-policy mldp1
   mdt c-multicast-routing bgp
   interface Loopback1
    enable
   !
   ssm range ssm1
  !
 !
!
```

Routers are ready to build their own P2MP trees.

```
RP/0/0/CPU0:SP2R3#sh mpls mldp bindings
mLDP MPLS Bindings database

LSP-ID: 0x00002 Paths: 1 Flags:
 0x00002 P2MP   12.39.3.3 [global-id 1]
   Local Label: 123309 Remote: 1048577 Inft: Lmdtsitee RPF-ID: 0 TIDv4/v6:
0xE0000011/0xE0800011

RP/0/0/CPU0:SP2R3#sh mpls mldp root
mLDP root database
 Root node     : 12.39.3.3  (We are the root)
  Metric       : 0
  Distance     : 0
  FEC count    : 1
  Path count   : 1
  Path(s)      : 12.39.3.3        LDP nbr: none


RP/0/0/CPU0:SP2R2#sh mpls mldp bindings
mLDP MPLS Bindings database

LSP-ID: 0x00002 Paths: 1 Flags:
 0x00002 P2MP   12.39.2.2 [global-id 1]
   Local Label: 123230 Remote: 1048577 Inft: Lmdtcust1 RPF-ID: 0 TIDv4/v6:
0xE0000012/0xE0800012

RP/0/0/CPU0:SP2R2#sh mpls mldp root
mLDP root database
 Root node     : 12.39.2.2  (We are the root)
  Metric       : 0
  Distance     : 0
  FEC count    : 1
  Path count   : 1
  Path(s)      : 12.39.2.2        LDP nbr: none
```

Now configure the BGP mVPN address family to enable autodiscovery and signalization for incoming PIM messages from the CE side.

Configure RR by copying neighbor settings from the VPNv4 AF.

```
SP2-R5-RR(config)#router bgp 1239
SP2-R5-RR(config-router)#add ipv4 mvpn
```

+ copy&paste the following settings from VPNv4:

```
SP2-R5-RR(config-router-af)#  neighbor 12.39.1.1 activate
SP2-R5-RR(config-router-af)#  neighbor 12.39.1.1 send-community extended
SP2-R5-RR(config-router-af)#  neighbor 12.39.1.1 route-reflector-client
SP2-R5-RR(config-router-af)#  neighbor 12.39.2.2 activate
SP2-R5-RR(config-router-af)#  neighbor 12.39.2.2 send-community extended
SP2-R5-RR(config-router-af)#  neighbor 12.39.2.2 route-reflector-client
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
SP2-R5-RR(config-router-af)#  neighbor 12.39.3.3 activate
SP2-R5-RR(config-router-af)#  neighbor 12.39.3.3 send-community extended
SP2-R5-RR(config-router-af)#  neighbor 12.39.3.3 route-reflector-client
SP2-R5-RR(config-router-af)#  neighbor 12.39.4.4 activate
SP2-R5-RR(config-router-af)#  neighbor 12.39.4.4 send-community extended
SP2-R5-RR(config-router-af)#  neighbor 12.39.4.4 route-reflector-client
```

Configure R1, R2. The mVPN AF statement under the VRF includes this VRF into the BGP advertisement.

```
!
router bgp 1239
 address-family ipv4 mvpn
 !
 neighbor 12.39.5.5
  address-family ipv4 mvpn
  !
 !
 vrf cust1
  address-family ipv4 mvpn
  !
 !
!
```

Configure R2 as follows.

```
!
router bgp 1239
 address-family ipv4 mvpn
 !
 neighbor 12.39.5.5
  address-family ipv4 mvpn
  !
 !
 vrf sitee
  address-family ipv4 mvpn
  !
 !
!
```

Verify the BGP signalization.

```
SP2-R5-RR#sh ip bgp ipv4 mvpn all summ
BGP router identifier 12.39.5.5, local AS number 1239
BGP table version is 12, main routing table version 12
4 network entries using 1216 bytes of memory
5 path entries using 640 bytes of memory
5/4 BGP path/bestpath attribute entries using 1440 bytes of memory
3 BGP AS-PATH entries using 88 bytes of memory
1 BGP community entries using 24 bytes of memory
29 BGP extended community entries using 3134 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 6542 total bytes of memory
BGP activity 248/182 prefixes, 2331469/2331378 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
12.39.1.1       4         1239      16     800       12    0    0 00:04:47        2
12.39.2.2       4         1239     776     809       12    0    0 00:04:45        2
12.39.3.3       4         1239      21     789       12    0    0 00:04:37        1
12.39.4.4       4         1239       0       0        1    0    0 never    (NoNeg)
```

There is a type-7 route which is the (S,G) entry signalled over BGP to all neighbors. As there are two requests for the mcast stream there are two BGP next hops 12.39.1.1 and 12.39.2.2 where the receivers are located.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
SP2-R5-RR#sh ip bgp ipv4 mvpn all
BGP table version is 12, local router ID is 12.39.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1
 *>i  [1][200:1][12.39.2.2]/12
                      12.39.2.2                100      0 i
Route Distinguisher: 200:2
 *>i  [1][200:2][12.39.1.1]/12
                      12.39.1.1                100      0 i
Route Distinguisher: 200:3
 *>i  [1][200:3][12.39.3.3]/12
                      12.39.3.3                100      0 i
 * i  [7][200:3][1239][192.168.3.31/32][238.1.1.1/32]/22
                      12.39.2.2                100      0 i
 *>i                   12.39.1.1               100       0 i
```

The BGP table on R3.

```
RP/0/0/CPU0:SP2R3#sh bgp ipv4 mvpn
BGP router identifier 12.39.3.3, local AS number 1239
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 9
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1
*>i[1][12.39.2.2]/40  12.39.2.2                100      0 i
Route Distinguisher: 200:2
*>i[1][12.39.1.1]/40  12.39.1.1                100      0 i
Route Distinguisher: 200:3 (default for vrf sitee)
*>i[1][12.39.1.1]/40  12.39.1.1                100      0 i
*>i[1][12.39.2.2]/40  12.39.2.2                100      0 i
*> [1][12.39.3.3]/40  0.0.0.0                             0 i
*>i[7][200:3][1239][32][192.168.3.31][32][238.1.1.1]/184
                      12.39.1.1                100      0 i

Processed 6 prefixes, 6 paths
```

The following command shows how to get more detailed information about the mVPN prefix.

```
RP/0/0/CPU0:SP2R3#
#sh bgp ipv4 mvpn rd 200:3 [7][200:3][1239][32][192.168.3.31][32][238.1.1.1]/184

BGP routing table entry for [7][200:3][1239][32][192.168.3.31][32][238.1.1.1]/184, Route
Distinguisher: 200:3
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 7           7
Last Modified: May 19 03:31:54.672 for 00:13:59
Paths: (1 available, best #1)
  Not advertised to any peer
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    12.39.1.1 (metric 20) from 12.39.5.5 (12.39.1.1)
      Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
imported
      Received Path ID 0, Local Path ID 0, version 7
      Extended community: RT:12.39.3.3:17
      Originator: 12.39.1.1, Cluster list: 12.39.5.5
      Source AFI: IPv4 MVPN, Source VRF: sitee, Source Route Distinguisher: 200:3
```

Based on this information the mLDP process can build P2MP trees. There are 3 P2MP trees. The R3 is the root for LSP-ID 0x0002. Outgoing labels are 123111 and 123231.

```
RP/0/0/CPU0:SP2R3#sh mpls mldp bindings
mLDP MPLS Bindings database

LSP-ID: 0x00002 Paths: 3 Flags:
 0x00002 P2MP   12.39.3.3 [global-id 1]
   Local Label: 123309 Remote: 1048577 Inft: Lmdtsitee RPF-ID: 0 TIDv4/v6:
0xE0000011/0xE0800011
   Remote Label: 123111 NH: 12.39.13.1 Inft: GigabitEthernet0/0/0/6
   Remote Label: 123231 NH: 12.39.23.2 Inft: GigabitEthernet0/0/0/0

LSP-ID: 0x00003 Paths: 2 Flags: Pk
 0x00003 P2MP   12.39.1.1 [global-id 1]
   Local Label: 123307 Active
   Remote Label: 1048577 Inft: Lmdtsitee RPF-ID: 1 TIDv4/v6: 0xE0000011/0xE0800011

LSP-ID: 0x00004 Paths: 2 Flags: Pk
 0x00004 P2MP   12.39.2.2 [global-id 1]
   Local Label: 123322 Active
   Remote Label: 1048577 Inft: Lmdtsitee RPF-ID: 5 TIDv4/v6: 0xE0000011/0xE0800011
```

Test the multicast stream.

```
RP/0/0/CPU0:SP2R3#ping vrf sitee 238.1.1.1 sou lo1 re 5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 238.1.1.1, timeout is 2 seconds:

Reply to request 0 from 21.41.16.1, 1 ms
Reply to request 0 from 192.168.7.7, 1 ms
Reply to request 1 from 21.41.16.1, 1 ms
Reply to request 1 from 192.168.7.7, 9 ms
Reply to request 2 from 21.41.16.1, 1 ms
Reply to request 2 from 192.168.7.7, 9 ms
Reply to request 3 from 192.168.7.7, 9 ms
Reply to request 3 from 21.41.16.1, 9 ms
Reply to request 4 from 21.41.16.1, 1 ms
Reply to request 4 from 192.168.7.7, 9 ms
```

You can capture the traffic on CER7 to verify the packet delivery.

```
CER7#sh mon cap cap buffer brief | i 192.168.3.31
 130   114     8.630024   192.168.3.31     ->  238.1.1.1         0  BE   ICMP
 166   114    10.951998   192.168.3.31     ->  238.1.1.1         0  BE   ICMP
 206   114    13.331983   192.168.3.31     ->  238.1.1.1         0  BE   ICMP
 239   114    15.579016   192.168.3.31     ->  238.1.1.1         0  BE   ICMP
 272   114    17.882010   192.168.3.31     ->  238.1.1.1         0  BE   ICMP
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.4: Inter-AS mVPN

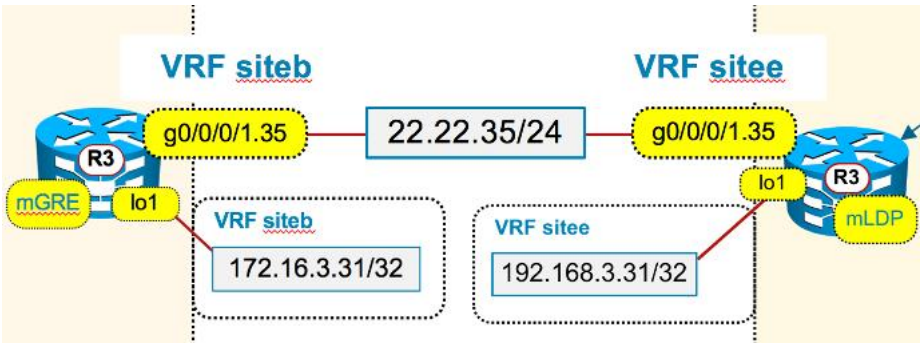
Connect both AS3356 and AS1239 and configure the multicast delivery from the VRF sitee 192.168.3.31 to 4 receivers:

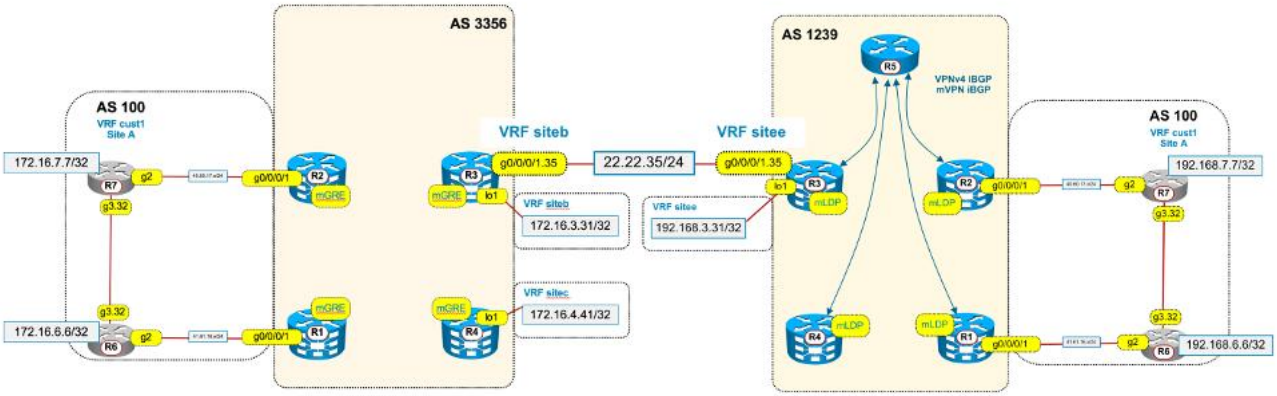
R7 in AS 100 - IP address: 172.16.7.7

R3 VRF siteb - IP address: 22.22.35.1

R7 in AS 200 - IP address: 192.168.7.7

R4 VRF cust1 - IP address: 21.41.16.1





R3:

```
!
interface GigabitEthernet0/0/0/1.35
 vrf sitee
 ipv4 address 22.22.35.2 255.255.255.0
 encapsulation dot1q 35
!
router pim
 vrf sitee
  address-family ipv4
   interface GigabitEthernet0/0/0/1.35
    enable
   !
  !
 !
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The PIM neighborship is up.

```
RP/0/0/CPU0:SP1R3#sh pim vrf siteb nei
[..]
Neighbor Address              Interface              Uptime    Expires  DR pri   Flags

22.22.35.1*                   GigabitEthernet0/0/0/1.35 06:51:59  00:01:41 1       B E
22.22.35.2                    GigabitEthernet0/0/0/1.35 00:00:46  00:01:22 1 (DR) B
```

There is no visibility of RP on SP2R3.

```
RP/0/0/CPU0:SP2R3#sh pim vrf sitee rp mapping
PIM Group-to-RP Mappings
```

But you don't need it if you use SSM. Configure routers in AS3356 to have an ability to join (S,G) directly.

R3:

```
!
ipv4 access-list mcast-238
 10 permit ipv4 host 238.1.1.1 any
!
router pim
 vrf siteb
  address-family ipv4
   ssm range mcast-238
  !
 !
!
```

R2:

```
!
ipv4 access-list mcast-238
 10 permit ipv4 host 238.1.1.1 any
!
router pim
 vrf cust1
  address-family ipv4
   ssm range mcast-238
  !
 !
!
```

R7:

```
!
ip igmp ssm-map enable
ip pim ssm range mcast-238
ip access-list standard mcast-238
 permit 238.1.1.1
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Join the multicast group 238.1.1.1 in AS3356.

```
CEL7(config)#int lo0
CEL7(config-if)#ip igmp join-group 238.1.1.1 source 192.168.3.31

RP/0/0/CPU0:SP1R3(config)#router igmp
RP/0/0/CPU0:SP1R3(config-igmp)#vrf siteb
RP/0/0/CPU0:SP1R3(config-igmp-siteb)#interface GigabitEthernet0/0/0/1.35
RP/0/0/CPU0:SP1R3(config-igmp-siteb-if)#join-group 238.1.1.1 192.168.3.31
RP/0/0/CPU0:SP1R3(config-igmp-siteb-if)#version 3
```

You need to set the correct routing to the source 192.168.3.31 on SP1R3.

R3:

```
!
router static
 vrf siteb
  address-family ipv4 unicast
   192.168.3.31/32 22.22.35.2
  !
 !
!
```

By default, the preference to 192.168.3.31 on R2 can be via R4. It must be through R3 as there is your SSM path.

```
RP/0/0/CPU0:SP1R2#sh route vrf cust1 | i 192.168.3.31
[..]
B    192.168.3.31/32 [200/0] via 33.56.4.4 (nexthop in vrf default), 00:00:12
```

To make the correction you can set the higher local preference so RR will choose the path via R3.

```
RP/0/0/CPU0:SP1R2#sh route vrf cust1 | i 192.168.3.31
[..]
B    192.168.3.31/32 [200/0] via 33.56.3.3 (nexthop in vrf default), 00:02:57
```

An example of the route policy modification.

R3:

```
!
route-policy LP200
  set local-preference 200
end-policy
!
router bgp 3356
 vrf siteb
  address-family ipv4 unicast
   redistribute static route-policy LP200
!
```

Please note that with this change the preference of the path to 192.168.3.31/32 in AS 1239 can be affected. Change the routing policy in a similar way in AS1239 so SP2R1 and SP2R2 take the path through SP2R3 not SP2R4.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

The expected result is as follows.

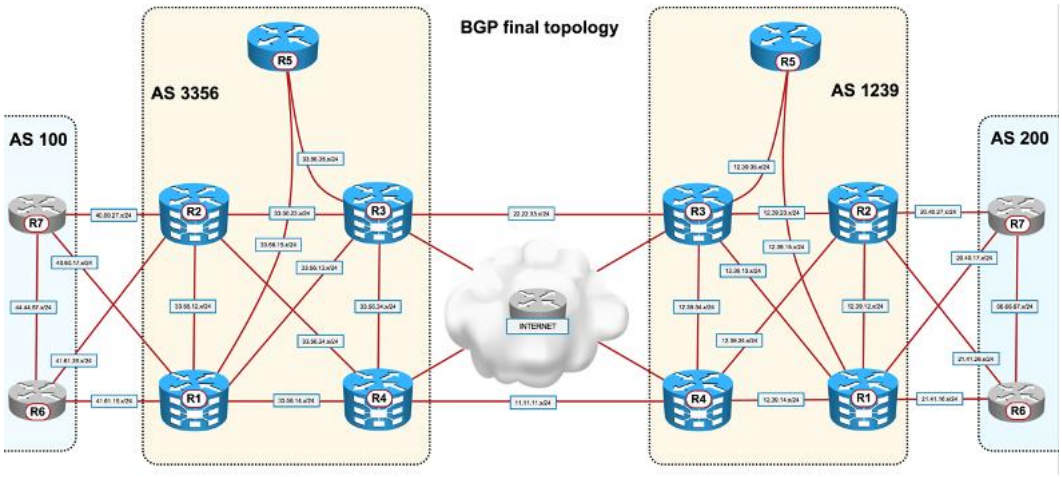
```
RP/0/0/CPU0:SP2R3#ping vrf sitee 238.1.1.1 sou lo1 re 5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 238.1.1.1, timeout is 2 seconds:

Reply to request 0 from 22.22.35.1, 1 ms
Reply to request 0 from 21.41.16.1, 1 ms
Reply to request 0 from 192.168.7.7, 1 ms
Reply to request 0 from 172.16.7.7, 9 ms
Reply to request 1 from 22.22.35.1, 9 ms
Reply to request 1 from 21.41.16.1, 9 ms
Reply to request 1 from 192.168.7.7, 9 ms
Reply to request 1 from 172.16.7.7, 9 ms
Reply to request 2 from 22.22.35.1, 9 ms
Reply to request 2 from 21.41.16.1, 9 ms
Reply to request 2 from 192.168.7.7, 9 ms
Reply to request 2 from 172.16.7.7, 9 ms
Reply to request 3 from 21.41.16.1, 9 ms
Reply to request 3 from 22.22.35.1, 9 ms
Reply to request 3 from 192.168.7.7, 19 ms
Reply to request 3 from 172.16.7.7, 19 ms
Reply to request 4 from 22.22.35.1, 1 ms
Reply to request 4 from 21.41.16.1, 1 ms
Reply to request 4 from 192.168.7.7, 1 ms
Reply to request 4 from 172.16.7.7, 9 ms
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 17: System management tasks

For the system tasks we'll use base topology you've build during Session 4:



This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch17.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch17.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch17.unl`

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: Logging functions

Task 1.1: Turning on logging

Both IOS-XE and IOS XR have very extensive instrumentation for logging. Compare the outputs:

```
CEL6(config)#logging ?
  Hostname or A.B.C.D  IP address of the logging host
  alarm                Configure syslog for alarms
  buffered             Set buffered logging parameters
  buginf               Enable buginf logging for debugging
  cns-events           Set CNS Event logging level
  console              Set console logging parameters
  count                Count every log message and timestamp last occurance
  delimiter            Append delimiter to syslog messages
  discriminator        Create or modify a message discriminator
  dmvpn                DMVPN Configuration
  esm                  Set ESM filter restrictions
  exception            Limit size of exception flush output
  facility             Facility parameter for syslog messages
  filter               Specify logging filter
  history              Configure syslog history table
  host                 Set syslog server IP address and parameters
  message-counter      Configure log message to include certain counter value
  monitor              Set terminal line (monitor) logging parameters
  on                   Enable logging to all enabled destinations
  origin-id            Add origin ID to syslog messages
  persistent           Set persistent logging parameters
  queue-limit          Set logger message queue size
  rate-limit           Set messages per second limit
  reload               Set reload logging level
  server-arp           Enable sending ARP requests for syslog servers when
                       first configured
  snmp-trap            Set syslog level for sending snmp trap
  source-interface     Specify interface for source address in logging
                       transactions
  trap                 Set syslog server logging level
  userinfo             Enable logging of user info on privileged mode enabling
```

With that of IOS XR:

```
RP/0/0/CPU0:SP1R1(config)#logging ?
  A.B.C.D or X:X::X  IP v4/v6 address of the logging host
  WORD               Name of the logging host
  archive            logging to a persistent device(disk/harddisk)
  buffered           Set buffered logging parameters
  console            Set console logging
  correlator         Configure properties of the event correlator
  events             Configure event monitoring parameters
  facility           Modify message logging facilities
  file               Set file logging
  history            Set history logging
  hostnameprefix     Hostname prefix to add on msgs to servers
  ipv4               Mark the dscp/precedence bit for ipv4 packets
  ipv6               Mark the dscp/precedence bit for ipv6 packets
  localfilesize      Set size of the local log file
  monitor            Set monitor logging
  source-interface   Specify interface for source address in logging transactions
  suppress           Configure properties for the event suppression
  suppress           Suppress logging behaviour
  trap               Set trap logging
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.1.1: Local buffered logging

First of all, we'll define local logging buffer and level of the logging itself. IOS-XE contrary to IOS XR gives us logging level cheat sheet - so we can check that debugging is the highest possible level (logging almost everything), while for example notifications is less detailed:

```
CEL6(config)#logging buffered ?
  <0-7>               Logging severity level
  <4096-2147483647>   Logging buffer size
  alerts              Immediate action needed           (severity=1)
  critical            Critical conditions               (severity=2)
  debugging           Debugging messages                (severity=7)
  discriminator       Establish MD-Buffer association
  emergencies         System is unusable                (severity=0)
  errors              Error conditions                  (severity=3)
  filtered            Enable filtered logging
  informational       Informational messages            (severity=6)
  notifications       Normal but significant conditions (severity=5)
  warnings            Warning conditions                (severity=4)
  xml                 Enable logging in XML to XML logging buffer
```

At the same time, IOS XR doesn't have this cheat sheet, and lists only names of the levels for logging:

```
RP/0/0/CPU0:SP1R1(config)#logging buffered ?
  <2097152-125000000>  Logging buffer size
  alerts               Immediate action needed
  critical             Critical conditions
  debugging            Debugging messages
  discriminator        Set logging buffer discriminator
  emergencies          System is unusable
  errors               Error conditions
  informational        Informational messages
  notifications        Normal but significant conditions
  warnings             Warning conditions
```

Let's configure logging buffer of 2MB and logging level of informational on both devices:

```
RP/0/0/CPU0:SP1R1(config)#logging buffered informational
RP/0/0/CPU0:SP1R1(config)#logging buffered 2097152 ! for IOS XR minimal value is
slightly
                                                   ! larger than 2MB
```

For IOS-XE the minimum size is just 4kB - it's legacy from very old IOS. Anyway:

```
CEL6(config)#logging buffered 2048000 informational
```

Let's check the setting on IOS XR:

```
RP/0/0/CPU0:SP1R1#sh logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: level warnings, 20 messages logged
    Monitor logging: level debugging, 0 messages logged
    Trap logging: level informational, 0 messages logged
    Buffer logging: level informational, 102 messages logged

Log Buffer (2097152 bytes):

RP/0/0/CPU0:May 20 22:21:49.526 : sysmgr[75]: %OS-SYSMGR-5-NOTICE : Card is COLD started
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RP/0/0/CPU0:May 20 22:21:53.976 : sysmgr[392]: %OS-SYSMGR-6-INFO : Backup system manager
is ready
```

...and on IOS-XE:

```
CEL6#sh logging
Syslog logging: enabled (0 messages dropped, 22 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)

No Active Message Discriminator.
No Inactive Message Discriminator.

    Console logging: level debugging, 88 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level informational, 2 messages logged, xml disabled,
                     filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled

No active filter modules.

    Trap logging: level informational, 109 message lines logged
        Logging Source-Interface:       VRF Name:

Log Buffer (2048000 bytes):

*Jan 01 08:37:47.294: %SYS-5-LOG_CONFIG_CHANGE: Buffer logging: level informational, xml
disabled, filtering disabled, size (2048000)
*Jan 01 08:38:34.450: %SYS-5-CONFIG_I: Configured from console by console
```

Task 1.1.2: Terminal (monitor) logging

Another often used and customized logging is the one that hits your terminal - also called monitor for legacy reasons. On IOS-XE it's configured via:

```
CEL6(config)#logging monitor debugging
```

On IOS XR with:

```
RP/0/0/CPU0:SP1R1(config)#logging monitor debugging
RP/0/0/CPU0:SP1R1(config)#commit
```

To designate (for this session only) specific terminal as the one that should also receive terminal logging information, on IOS-XE you use:

```
CEL6#terminal monitor
% Console already monitors ! we're logged over console – so it already monitors
CEL6#
```

On IOS XR:

```
RP/0/0/CPU0:SP1R1#terminal monitor
terminal monitor error:console already monitors ! we're logged over console
RP/0/0/CPU0:SP1R1#
```

While we're at logging, there's nice trick to make every log on the console properly terminate with new line and get us to the prompt position we were on

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

already. While on IOS XR this feature is not yet available (tracked by **CSCs138138**), it's on IOS-XE:

```
CEL6(config)#line console 0
CEL6(config-line)#logging synchronous
CEL6(config-line)#line vty 0 15
CEL6(config-line)#logging synchronous
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.1.3: Console logging

Separate configuration option is available also for console connection. This may make sense for specific configurations, where for example console is connected all of the time via terminal server for example. On IOS-XE configuration is pretty straight forward:

```
CEL6(config)#logging console debugging
*Jan 01 08:50:49.335: %SYS-5-CONFIG_I: Configured from console by console[OK]
```

On IOS XR with (You'll immediately show debug message about the commit itself):

```
RP/0/0/CPU0:SP1R1(config)#logging console debugging
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
RP/0/0/CPU0:Jan 01 08:50:29.272 : config[65744]: %MGBL-CONFIG-6-DB_COMMIT :
Configuration committed by user 'cisco'. Use 'show configuration commit changes
1000000004' to view the changes.
RP/0/0/CPU0:Jan 01 08:50:29.292 : config[65744]: %MGBL-SYS-5-CONFIG_I : Configured from
console by cisco
```

It’s good practice to standardize whole organization on one way of measuring and logging time, and then use it. In case of any trouble, with NTP properly running, You’ll have one set of timestamps to compare:

```
CEL6(config)#service timestamps log datetime localtime msec show-timezone
CEL6(config)#service timestamps debug datetime localtime msec show-timezone
```

Format will change:

```
*Jan 01 21:40:04.419: %BGP-5-ADJCHANGE: […]
*Jan 01 22:07:43.782 UTC: %SYS-5-CONFIG_I: […]
```

As with IOS XR:

```
RP/0/0/CPU0:SP1R1(config)#service timestamps log datetime localtime msec show-timezone
RP/0/0/CPU0:SP1R1(config)#service timestamps debug datetime localtime msec show-timezone
```

The format slightly changes, as by default, IOS XR already logs date and time:

```
RP/0/0/CPU0:Jan 01 22:02:21.706 : exec[65701]: […]
RP/0/0/CPU0:Jan 01 22:06:14.400 UTC: config[65743]: […]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.1.4: Logging filters

Another useful feature is ability to filter out totally or rate limit specific messages, facility levels or severities. Let's filter out in the memory buffer ALL messages that contain LINEPROTO:

```
CEL6(config)#logging discriminator DROP_INT msg-body drops LINEPROTO
CEL6(config)#logging buffered discriminator DROP_INT
```

For IOS XR the logging features are more extensive, and can group messages to filter using complex policies to suppress/unsuppress or set thresholds for specific messages. To look up categories, groups and message codes to apply policies to, you can always check the column names using:

```
RP/0/0/CPU0:SP1R2#show logging events buffer all-in-buffer
#ID    :C_id:Source  :Time                :%GROUP-SEVERITY-MESSAGECODE: Text
#1     :    :RP/0/0/CPU0:Jan 01 21:34:47.409 2017:init[65549]: %OS-INIT-7-MBI_STARTED :
total time 1.589 seconds
```

To check messages of specific type while working interactively, you can use category filter, like this:

```
RP/0/0/CPU0:SP1R2#show logging events buffer category SECURITY
#ID    :C_id:Source  :Time                :%GROUP-SEVERITY-MESSAGECODE: Text
#8     :    :RP/0/0/CPU0:Jan 01 21:36:09.034 2017:sam_server[362]: %SECURITY-SAM-6-
INFO_INSECURE_FLASH_FILE : /disk0/sam_certdb is detected insecure, backup not restored
#9     :    :RP/0/0/CPU0:Jan 01 21:36:10.054 2017:sam_server[362]: %SECURITY-SAM-3-
ERROR_2_PARAM : Failed setting I_BIT on backup file, /disk0/sam_certdb
#11    :    :RP/0/0/CPU0:Jan 01 21:36:20.793 2017:sam_server[362]: %SECURITY-SAM-4-
WARNING : Failed to initialize nvram digest
#15    :    :RP/0/0/CPU0:Jan 01 21:37:44.577 2017:sam_server[362]: %SECURITY-SAM-4-
WARNING : Failed to initialize nvram digest
#17    :    :RP/0/0/CPU0:Jan 01 21:38:56.092 2017:ipsec_mp[232]: %SECURITY-IMP-6-
PROC_READY : Process ipsec_mp is ready
#75    :    :RP/0/0/CPU0:Jan 01 18:38:19.067 2017:exec[65701]: %SECURITY-LOGIN-6-
AUTHEN_SUCCESS : Successfully authenticated user 'cisco' from 'console' on 'con0_0_CPU0'
```

This can be handy while debugging or troubleshooting some scenario.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.1.5: Supressing duplicate messages – IOS XR only

Additionally, IOS XR has a feature to suppress duplicates – for the next 60, 120 and then 180 seconds (by default), messages are only counted and printed as summary when they repeat themselves. You activate this feature by committing following command:

```
RP/0/0/CPU0:SP1R2(config)#logging suppress duplicates
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: Logging to external syslog server

In IOS XR, logging to external Syslog server needs either default severity definition, or at least defining port – for default (514) or some specific number:

```
RP/0/0/CPU0:SP1R2(config)#logging 10.6.6.10 port default
RP/0/0/CPU0:SP1R2(config)#commit
```

On IOS-XE the basic configuration is very similar:

```
CEL6(config)#logging host 10.6.6.10
```

You can define discriminator as for console/terminal/buffered logging:

```
CEL6(config)#logging host 10.6.6.10 discriminator DROP_INT
```

This can be then verified using ‘sh logging’:

```
CEL6#sh logging
Syslog logging: enabled (0 messages dropped, 22 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)

Active Message Discriminator:
DROP_INT  msg-body       drops    LINEPROTO

No Inactive Message Discriminator.

[…]

No active filter modules.

    Trap logging: level informational, 115 message lines logged
        Logging to 10.6.6.10  (udp port 514, audit disabled,
              link up),
              3 message lines logged,
              0 message lines rate-limited,
              0 message lines dropped-by-MD,
              xml disabled, sequence number disabled
              filtering enabled, discriminator (DROP_INT)
        Logging Source-Interface:       VRF Name:
```

Task 1.2.1: Assigning specific prefix/name for logging host

Sometimes organizations standardize on different naming scheme for logging vs naming of the nodes. In such situation you can add specific prefix for logging messages.

On IOS-XE, there are couple of predefined values or option to define your own logging prefix using ‘string’ parameter. Let’s set the prefix to SP1CE7 on CEL7:

```
CEL7(config)#logging origin-id ?
  hostname  Use origin hostname as ID
  ip        Use origin IP address as ID
  ipv6      Use origin IPv6 address as ID
  string    Define a unique text string as ID
  <cr>
CEL7(config)#logging origin-id string SP1CE7
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

On SP1R2 let's set the prefix to SP1R2-EDGE:

```
RP/0/0/CPU0:SP1R2(config)#logging hostnameprefix SP1R2-EDGE
RP/0/0/CPU0:SP1R2(config)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3: Logging security

For typical use cases, enabling sequence numbers on generated logs serves two purposes at the same time – security and troubleshooting. If we're seeing gap in the sequenced logs, it's either because of some tampering of in-flight packets (like syslog messages) or problem with delivery. Anyway, apart from keeping the time current and synchronized across the network, turning on sequence numbers for logs is best practice. On IOS-XE it's done this way:

```
CEL6(config)#logging host 3.3.3.3 sequence-num-session
CEL6#sh log | b 3.3.3.3
        Logging to 3.3.3.3  (udp port 514, audit disabled,
              link up),
              3 message lines logged,
              0 message lines rate-limited,
              0 message lines dropped-by-MD,
              xml disabled, sequence number enabled
              filtering disabled
        Logging Source-Interface:       VRF Name:
```

To turn the sequence numbers globally, for both syslog and memory logging, enable service:

```
CEL6(config)#service sequence-numbers
```

On IOS XR sequence numbers are not available.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4: SNMP configuration

There are three versions of the SNMP protocol, with SNMPv1 and SNMPv2c ("community" vs SNMPv2p – "port-based" – it's about security model) offering only basic control with no encryption of communication, and SNMPv3 offering strong control, authentication and encryption.

Let's configure basic SNMP configuration for IOS-XE and IOS XR:

Task 1.4.1: Basic SNMPv2c configuration

Let's configure initial SNMP information:

```
CEL6(config)#snmp-server chassis-id CEL6 ! node identifier
CEL6(config)#snmp-server ifindex persist ! no ifIndex changes between reboots
CEL6(config)#snmp-server location SP-BCPL ! location – good practice
```

SNMP has views concept, and you define and include views to control which communities (SNMPv2c) or users and groups (SNMPv3) can access which information and in which mode (read and/or write). By default on IOS-XE there are three views defined:

```
CEL6#sh snmp view
*ilmi system - included permanent active
*ilmi atmForumUni - included permanent active
cac_view pimMIB - included read-only active
[…]
v1default iso - included permanent active
v1default internet - included permanent active
[…]
```

Let's configure basic community-based access to read-only view only to 'interfaces' MIB, without ACLs controlling access (this is against best practices of course!):

```
CEL6(config)#snmp-server community COMM ?
  <1-99>        Std IP accesslist allowing access with this community string
  <1300-1999>   Expanded IP accesslist allowing access with this community
                string
  WORD          Access-list name
  ipv6          Specify IPv6 Named Access-List
  ro            Read-only access with this community string
  rw            Read-write access with this community string
  view          Restrict this community to a named MIB view
  <cr>
CEL6(config)#snmp-server community COMM view interfaces ro
```

SNMP can also generate traps and informs to remote servers:

```
CEL6(config)#snmp-server host 12.39.1.1 traps version 2c COMM
CEL6(config)#snmp-server host 12.39.1.1 informs version 2c COMM
```

Communication from SNMP can be then verified with:

```
CEL6#sh snmp
Chassis: CEL6
Location: SP-BCPL
0 SNMP packets input
[…]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
0 SNMP packets output
[…]
SNMP global trap: disabled

SNMP logging: enabled
    Logging to 12.39.1.1.162, 0/10, 0 sent, 0 dropped.

SNMP Manager-role output packets
[…]
SNMP Manager-role input packets
[…]

SNMP informs: enabled
    Informs in flight 0/25 (current/max)
    Logging to 12.39.1.1.162
        0 sent, 0 in-flight, 0 retries, 0 failed, 0 dropped
```

Let's enable traps:

```
CEL6(config)#snmp-server enable traps snmp linkdown linkup
CEL6(config)#snmp-server enable traps ospf
CEL6(config)#snmp-server enable traps bgp
CEL6(config)#snmp-server enable traps ipsla
CEL6(config)#snmp-server enable traps config-copy
CEL6(config)#snmp-server enable traps config
CEL6(config)#snmp-server enable traps mpls ldp
CEL6(config)#snmp-server enable traps alarms informational
```

As soon as SNMP will register specific events, it will send packets to defined server. As we don't have SNMP logging host, we can either monitor counters or packet captures. Let's check counters:

```
CEL6#sh snmp
Chassis: CEL6
Location: SP-BCPL
0 SNMP packets input
[…]
26 SNMP packets output
[…]
    11 Trap PDUs
SNMP global trap: disabled

SNMP logging: enabled
    Logging to 12.39.1.1.162, 1/10, 10 sent, 0 dropped.

SNMP Manager-role output packets
[…]
    15 Inform-request PDUs
    4 Timeouts
    0 Drops
[…]
SNMP informs: enabled
    Informs in flight 11/25 (current/max)
    Logging to 12.39.1.1.162
        11 sent, 11 in-flight, 0 retries, 0 failed, 0 dropped
```

You can also display SNMP host-related information (based on the configuration we've did above):

```
CEL6#sh snmp host
Notification host: 12.39.1.1    udp-port: 162   type: inform
user: COMM      security model: v2c

Notification host: 12.39.1.1    udp-port: 162   type: trap
user: COMM      security model: v2c
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.5: RMON configuration

RMON is very simple engine to generate alarms on crossing some specific thresholds.

```
CEL6(config)#rmon event 10 ?
  description  Specify a description of the event
  log          Generate RMON log when the event fires
  owner        Specify an owner for the event
  trap         Generate SNMP trap when the event fires
```

RMON can generate SNMP trap or RMON log information. Let's define specific log entry:

```
CEL6(config)#rmon event 10 log description TEST-RMON
```

Each RMON event configuration is kept in the RMON events database:

```
CEL6#sh rmon events
Event 10 is active, owned by config
 Description is TEST-RMON
 Event firing causes log,
 last event fired at  0y0w0d,00:00:00,
 Current uptime       0y0w0d,00:53:39
```

Let's define simple alarm for Gi3 interface. To do that, you need to enable basic SNMP configuration:

```
CEL6(config)#snmp-server community SNMP-T
```

Now, we're going to check the interface number according to SNMP and use it in the next RMON configuration:

```
CEL6#show snmp mib ifmib ifindex
GigabitEthernet5: Ifindex = 5
GigabitEthernet1: Ifindex = 1
GigabitEthernet3: Ifindex = 3
GigabitEthernet3-mpls layer: Ifindex = 9
VoIP-Null0: Ifindex = 6
Loopback0: Ifindex = 8
Null0: Ifindex = 7
GigabitEthernet4: Ifindex = 4
GigabitEthernet2: Ifindex = 2
```

So, we're going to use interface 3 to check for the bandwidth usage (changes) on the interface. We'll define two additional probes, checking for both:

```
CEL6(config)#rmon alarm 10 ifSpeed.3 5 absolute rising-threshold 10 falling-threshold 5
```

The RMON alarms registers the configuration:

```
CEL6#sh rmon alarms
Total number of alarm trap failures 0
Alarm 10 is active, owned by config
 Monitors ifSpeed.3.3 every 10 second(s)
 Taking absolute samples, last value was 0
 Rising threshold is 100, assigned to event 1
 Falling threshold is 10, assigned to event 2
 On startup enable rising or falling alarm
 Alarm configured for interface GigabitEthernet3 with ifindex 3
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Also, the RMON events are visible once configured:

```
CEL6#sh rmon events
Event 1 is active, owned by config
 Description is Gi3-FALLING
 Event firing causes log,
 last event fired at  0y0w0d,00:00:00,
 Current uptime       0y0w0d,01:03:55
Event 2 is active, owned by config
 Description is Gi3-RAISING
 Event firing causes log,
 last event fired at  0y0w0d,00:00:00,
 Current uptime       0y0w0d,01:03:55
Event 10 is active, owned by config
 Description is TEST-RMON
 Event firing causes log,
 last event fired at  0y0w0d,00:00:00,
 Current uptime       0y0w0d,01:03:55
```

The interface will immediately raise an alarm after definition, as Gi3 speed is set to 1Gbit/s:

```
*Jan 01 20:46:59.576: %RMON-5-RISINGTRAP: Rising threshold has been crossed because the
value of ifSpeed.3 exceeded the rising-threshold value 10
```

Now, let's check what will happen if we downgrade bandwidth of interface:

```
CEL6(config)#int gi3
CEL6(config-if)#bandwidth 500
```

RMON will raise an alarm:

```
*Jan 01 20:50:09.097: %RMON-5-FALLINGTRAP: Falling threshold has been crossed because
the value of ifSpeed.3 has fallen below the falling-threshold value 5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Debugging network traffic using captures

Task 2.1: EPC – IOS-XE

Embedded Packet Capture has a lot of different options to help you facilitate network troubleshooting.

Let's start with simple session and define capture source:

```
CEL6#monitor capture MyCAP ?
  WORD            Name of the Capture
  access-list     access-list to be attached
  buffer          Buffer options
  class-map       class name to attached
  clear           Clear Buffer
  control-plane   Control Plane
  export          Export Buffer
  interface       Interface
  limit           Limit Packets Captured
  match           Describe filters inline
  start           Enable Capture
  stop            Disable Capture
```

Source can be either ACL (access-list), defined earlier class-map matching some interesting traffic (class-map), interface or some specific filter defined from the command-line (match). Filters usually require definition of interface to attach to. Let's define filter for all IPv6 traffic on Gi2 in both directions (inbound and outbound):

```
CEL6(config)#ipv6 access-list CAP-T01
CEL6(config-ipv6-acl)#permit ipv6 any any
CEL6(config-ipv6-acl)#^Z
CEL6#monitor capture MyCAP access-list CAP-T01
CEL6#monitor capture MyCAP interface GigabitEthernet 2 both
```

By default, monitoring is turned off, with no limits for packet capture (with default upper limit of 10MB) and linear capture buffer:

```
CEL6#show monitor capture

Status Information for Capture MyCAP
  Target Type:
 Interface: GigabitEthernet2, Direction: BOTH
   Status : Inactive
  Filter Details:
   Access-list: CAP-T01
  Buffer Details:
   Buffer Type: LINEAR (default)
   Buffer Size (in MB): 10
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Maximum number of packets to capture per second: 1000
   Packet sampling rate: 0 (no sampling)
```

Let's start the monitoring and generate some IPv6 traffic:

```
CEL6#monitor capture MyCAP start
Started capture point : MyCAP
000121: *Jan 01 08:05:30.588 UTC: %BUFCAP-6-ENABLE: Capture Point MyCAP enabled.
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

We can interactively check buffer status:

```
CEL6#show monitor capture MyCAP buffer
 buffer size (KB) : 10240
 buffer used (KB) : 128
 packets in buf   : 5
 packets dropped  : 0
 packets per sec  : 0
```

Of course for troubleshooting we'll need to take a look into the buffer itself. 'brief' view shows just hosts exchanging the traffic:

```
CEL6#show monitor capture MyCAP buffer brief
 ----------------------------------------------------------------------------
 #   size   timestamp     source              destination      dscp    protocol
 ----------------------------------------------------------------------------
   0  118    0.000000    FE80:*:0006     ->   FF02:*:0001     56 CS7  IPv6-ICMP
   1  113   11.475033    2001:*:0001     ->   2001:*:0006     48 CS6  TCP
   2   94   11.676011    2001:*:0006     ->   2001:*:0001     48 CS6  TCP
   3  113   24.182989    2001:*:0006     ->   2001:*:0001     48 CS6  TCP
   4   94   24.404038    2001:*:0001     ->   2001:*:0006     48 CS6  TCP
   5   86   62.402979    FE80:*:0001     ->   2001:*:0006     56 CS7  IPv6-ICMP
   6   78   62.405029    2001:*:0006     ->   FE80:*:0001     56 CS7  IPv6-ICMP
```

'**dump**' allows as to see hex-dump of traffic:

```
CEL6#show monitor capture MyCAP buffer dump
0 ! this is captured packet number
  0000:  33330000 0001FA16 3EAE5C9E 86DD6E00   33......>.\...n.
  0010:  00000040 3AFFFE80 00000000 00000000   ...@:...........
  0020:  00000000 0006FF02 00000000 00000000   ................
  0030:  00000000 00018600 67824000 07080000   ........g.@.....
  0040:  00000000 00000101 FA163EAE 5C9E0501   ..........>.\...
  0050:  00000000 05DC0304 40C00027 8D000009   ........@..'....
  0060:  3A800000 00002001 00410061 00160000   :..... ..A.a....
  0070:  00000000 0000                         ......

1
  0000:  FA163EAE 5C9EFA16 3E3D52A9 86DD6C00   ..>.\...>=R...l.
  0010:  0000003B 06002001 00410061 00160000   ...;.. ..A.a....
  0020:  00000000 00012001 00410061 00160000   ...... ..A.a....
  0030:  00000000 000688E1 00B3B0CD 3DD1A026   ............=..&
  0040:  4E1FA018 7DEC973D 00001312 D37857FB   N...}..=.....xW.
  0050:  8309CBD0 A474FE74 0BA1628B 0000FFFF   .....t.t..b.....
  0060:  FFFFFFFF FFFFFFFF FFFFFFFF FFFF0013   ................
  0070:  04                                    .
```

While '**detailed**' is mix of both modes – shows traffic direction, QoS and protocol data, while also displaying the hex dump of packet contents:

```
CEL6#show monitor capture MyCAP buffer detailed
 ----------------------------------------------------------------------------
 #   size   timestamp     source              destination      dscp    protocol
 ----------------------------------------------------------------------------
   0  118    0.000000    FE80:*:0006     ->   FF02:*:0001     56 CS7  IPv6-ICMP
  0000:  33330000 0001FA16 3EAE5C9E 86DD6E00   33......>.\...n.
  0010:  00000040 3AFFFE80 00000000 00000000   ...@:...........
  0020:  00000000 0006FF02 00000000 00000000   ................
  0030:  00000000 00018600 67824000 07080000   ........g.@.....

   1  113   11.475033    2001:*:0001     ->   2001:*:0006     48 CS6  TCP
  0000:  FA163EAE 5C9EFA16 3E3D52A9 86DD6C00   ..>.\...>=R...l.
  0010:  0000003B 06002001 00410061 00160000   ...;.. ..A.a....
  0020:  00000000 00012001 00410061 00160000   ...... ..A.a....
  0030:  00000000 000688E1 00B3B0CD 3DD1A026   ............=..&
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
   2   94   11.676011    2001:*:0006     ->   2001:*:0001     48 CS6  TCP
  0000:  FA163E3D 52A9FA16 3EAE5C9E 86DD6C00   ..>=R...>.\...l.
  0010:  00000028 06012001 00410061 00160000   ...(.. ..A.a....
  0020:  00000000 00062001 00410061 00160000   ...... ..A.a....
  0030:  00000000 000100B3 88E1A026 4E1FB0CD   ...........&N...
```

Let's stop capture:

```
CEL6#monitor capture MyCAP stop
Stopped capture point : MyCAP
000122: *Jan 01 08:10:55.579 UTC: %BUFCAP-6-DISABLE: Capture Point MyCAP disabled.
CEL6#show monitor capture MyCAP buffer
 buffer size (KB) : 10240
 buffer used (KB) : 128
 packets in buf   : 37
 packets dropped  : 0
 packets per sec  : 0
```

We can now export captured traffic to remote destination or local filesystem. Let's do that – saving the capture to flash filesystem:

```
CEL6#monitor capture MyCAP export flash:MyCAP.pcap
Exported Successfully
CEL6#dir flash:*.pcap
Directory of bootflash:/*.pcap

Directory of bootflash:/

   14  -rw-        4247  Jan 01 2017 08:11:50 +00:00  MyCAP.pcap
7897796608 bytes total (6520393728 bytes free)
```

Let's now change the buffer characteristics – from linear to circular (overwrites oldest captured packets if the capture process reaches the end of the buffer), size of 1MB:

```
CEL6#monitor capture MyCAP buffer circular size 1
Buffer size is already set[Replace]?[confirm]
```

Now, let's change the capture definition to keep packets hitting control plane of router and remove interface from matching traffic:

```
CEL6#monitor capture MyCAP control-plane both
CEL6#no monitor capture MyCAP interface gigabitEthernet 2
CEL6#show monitor capture

Status Information for Capture MyCAP
  Target Type:
 Interface: Control Plane, Direction: BOTH
   Status : Inactive
  Filter Details:
   Access-list: CAP-T01
  Buffer Details:
   Buffer Type: CIRCULAR
   Buffer Size (in MB): 1
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Maximum number of packets to capture per second: 1000
   Packet sampling rate: 0 (no sampling)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

And start the capture:

```
CEL6#monitor capture MyCAP start
Started capture point : MyCAP
```

Given we're running multiple control plane protocols in the network, after waiting 60 seconds we should have already couple of packets in the buffer:

```
CEL6#show monitor capture MyCAP buffer
 buffer size (KB) : 1024
 buffer used (KB) : 128
 packets in buf   : 37
 packets dropped  : 0
 packets per sec  : 0
```

Stop the capture:

```
CEL6#monitor capture MyCAP stop
```

Now, check capture buffer contents:

```
CEL6#show monitor capture MyCAP buffer brief
 ----------------------------------------------------------------------------
 #   size   timestamp     source              destination      dscp    protocol
 ----------------------------------------------------------------------------
   0    94     0.000000    FE80:*:0006     ->   FF02:*:0005     48 CS6  OSPF
   1    94     2.313979    FE80:*:0007     ->   FF02:*:0005     48 CS6  OSPF
   2   118     3.831958    FE80:*:0006     ->   FF02:*:0001     56 CS7  IPv6-ICMP
   3    94     9.210988    FE80:*:0006     ->   FF02:*:0005     48 CS6  OSPF
   4    94    12.071988    FE80:*:0007     ->   FF02:*:0005     48 CS6  OSPF
   5    94    18.593964    FE80:*:0006     ->   FF02:*:0005     48 CS6  OSPF
   6    94    21.269991    FE80:*:0007     ->   FF02:*:0005     48 CS6  OSPF
   7    86    24.348981    FE80:*:0001     ->   2001:*:0006     56 CS7  IPv6-ICMP
   8    78    24.350980    2001:*:0006     ->   FE80:*:0001     56 CS7  IPv6-ICMP
   9    94    27.919951    FE80:*:0006     ->   FF02:*:0005     48 CS6  OSPF
  10   113    28.557970    2001:*:0006     ->   2001:*:0001     48 CS6  TCP
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3: Network telemetry

Task 3.1: Configuring NetFlow in IOS/IOS-XE

Modern versions of IOS and IOS-XE support only Flexible NetFlow (v9). This feature however can behave and export data compatible with v5 flow collectors using legacy definition of flows – for key fields (netflow and netflow-original) and export format.

For now, we'll focus on configuring native, Flexible NetFlow capabilities. Configuration happens in three steps: defining key fields (flow record), monitor (flow monitor) and export data specification (flow exporter).

Task 3.1.1: Defining flow key records

Key fields in v9 can be customized to contain specific sets of data – that was not possible with vanilla v5 exports. Let's check what are the options:

```
CEL6(config)#flow record NF-REC-NETWORK
CEL6(config-flow-record)#match ipv4 ?
  destination     IPv4 destination address based fields
  dscp            IPv4 DSCP (part of TOS)
  fragmentation   IPv4 fragmentation fields
  header-length   IPv4 header length (IHL in 32 bit words)
  id              IPv4 ID
  length          IPv4 length fields
  option          IPv4 option fields
  precedence      IPv4 precedence (part of TOS)
  protocol        IPv4 protocol
  section         Part of the packet
  source          IPv4 source address based fields
  tos             IPv4 type of service
  total-length    IPv4 total length
  ttl             IPv4 TTL
  version         IP version from IPv4 header
```

Let's assume that our flows will be created for every unique source/destination IPv4 pair:

```
CEL6(config-flow-record)#match ipv4 source address
CEL6(config-flow-record)#match ipv4 destination address
```

For that flows, we'll collect in flow export information following data (if NetFlow feature can find them in the packets qualified as unique flows):

```
CEL6(config-flow-record)#collect mpls label 1 details
CEL6(config-flow-record)#collect network delay sum
CEL6(config-flow-record)#collect ipv4 protocol
```

After defining the record, we can check it's configuration:

```
CEL6#show flow record NF-REC-NETWORK
flow record NF-REC-NETWORK:
  Description:        User defined
  No. of users:       0
  Total field space:  20 bytes
  Fields:
    match ipv4 source address
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
    match ipv4 destination address
    collect ipv4 protocol
    collect mpls label 1 details
    collect network delay sum
```

Task 3.1.2: Defining flow monitor

Now, we'll define a flow monitor that will use defined record flow definition to gather information for exporting:

```
CEL6(config)#flow monitor NF-MON-NETWORK
CEL6(config-flow-monitor)#record NF-REC-NETWORK
CEL6(config-flow-monitor)#^Z
CEL6#show flow monitor
Flow Monitor NF-MON-NETWORK:
  Description:       User defined
  Flow Record:       NF-REC-NETWORK
  Cache:
    Type:                normal (Platform cache)
    Status:              allocated
    Size:                200000 entries
    Inactive Timeout:    15 secs
    Active Timeout:      1800 secs
    Trans end aging:   off
```

Task 3.1.3: Defining flow exporter

The last part of definition will describe where and how we're going to export the data:

```
CEL6(config)#flow exporter NF-EXP-NETWORK
CEL6(config-flow-exporter)#description NFv9 export
CEL6(config-flow-exporter)#destination 12.39.5.5
CEL6(config-flow-exporter)#export-protocol ?
  ipfix       IPFIX (Version 10) ! notice you can export in multiple formats
  netflow-v5  NetFlow Version 5
  netflow-v9  NetFlow Version 9
CEL6(config-flow-exporter)#export-protocol netflow-v9
```

We need now to get back to flow monitor and attach the exporter to it:

```
CEL6(config)#flow monitor NF-MON-NETWORK
CEL6(config-flow-monitor)#exporter NF-EXP-NETWORK
```

Now the monitor looks like this:

```
CEL6#show flow monitor
Flow Monitor NF-MON-NETWORK:
  Description:       User defined
  Flow Record:       NF-REC-NETWORK
  Flow Exporter:     NF-EXP-NETWORK
  Cache:
    Type:                normal (Platform cache)
    Status:              allocated
    Size:                200000 entries
    Inactive Timeout:    15 secs
    Active Timeout:      1800 secs
    Trans end aging:   off
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

And flow exporter:

```
CEL6#show flow exporter
Flow Exporter NF-EXP-NETWORK:
  Description:              NFv9 export
  Export protocol:          NetFlow Version 9
  Transport Configuration:
    Destination IP address: 12.39.5.5
    Source IP address:      41.61.16.6
    Transport Protocol:     UDP
    Destination Port:       9995
    Source Port:            50565
    DSCP:                   0x0
    TTL:                    255
    Output Features:        Used
```

Task 3.1.4: Attaching NetFlow feature to interfaces

Now we can check if there's any traffic coming over CEL6:

```
CEL6(config)#interface gigabitEthernet 2
CEL6(config-if)#ip flow monitor NF-MON-NETWORK input
CEL6(config-if)#ip flow monitor NF-MON-NETWORK output
CEL6(config)#int gi3
CEL6(config-if)#ip flow monitor NF-MON-NETWORK input
CEL6(config-if)#ip flow monitor NF-MON-NETWORK output
CEL6(config-if)#int gi4
CEL6(config-if)#ip flow monitor NF-MON-NETWORK input
CEL6(config-if)#ip flow monitor NF-MON-NETWORK output
```

As soon as we'll assign the interfaces to monitoring, we'll start gathering the flow data in cache:

```
CEL6#show flow monitor NF-MON-NETWORK cache format table
  Cache type:                               Normal (Platform cache)
  Cache size:                               200000
  Current entries:                               9
  High Watermark:                                9

  Flows added:                                  15
  Flows aged:                                    6
    - Inactive timeout    (    15 secs)          6

IPV4 SRC ADDR    IPV4 DST ADDR    mpls label 1  app network delay sum  ip prot
===============  ===============  ============  =====================  =======
41.61.16.1       41.61.16.6          0      /0                      0        6
44.44.67.7       224.0.0.5           0      /0                      0       89
172.16.6.6       172.16.7.7          0      /0                      0        6
44.44.67.6       224.0.0.5           0      /0                      0       89
172.16.7.7       172.16.6.6          0      /0                      0        6
44.44.67.7       224.0.0.2           0      /0                      0       17
44.44.67.6       224.0.0.2           0      /0                      0       17
41.61.16.6       41.61.16.1          0      /0                      0        6
41.61.16.6       12.39.5.5           0      /0                      0       17
```

There's a lot of features available in parser to process the cache information, including the top-N feature.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Let’s sort IPv4 source addresses from highest and check top 10 flows in cache:

```
CEL6#show flow monitor NF-MON-NETWORK cache sort highest ipv4 source address top 10
Processed 7 flows
Aggregated to 7 flows
Showing the top 7 flows

IPV4 SRC ADDR    IPV4 DST ADDR    mpls label 1  app network delay sum  ip prot
===============  ===============  ============  =====================  =======
172.16.7.7       172.16.6.6          0      /0                      0        6
172.16.6.6       172.16.7.7          0      /0                      0        6
44.44.67.7       224.0.0.5           0      /0                      0       89
44.44.67.7       224.0.0.2           0      /0                      0       17
44.44.67.6       224.0.0.5           0      /0                      0       89
44.44.67.6       224.0.0.2           0      /0                      0       17
41.61.16.6       12.39.5.5           0      /0                      0       17
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2: BGP policy accounting in IOS/IOS-XE

BGP policy accounting enables to account for traffic based on BGP attributes. This way, instead of using simple interface counters, you can aggregate prefix counters based for example on communities, AS_PATH and others.

Before continuing, please check if you have working BGP configuration across AS 100, AS 3356, AS 1239 and AS 200 before configuring.

Let's assume, we want to count ALL traffic coming in from AS 200. Those are the prefixes currently in CEL7 BGP table:

```
CEL7#sh bgp ipv4 unicast regexp _200$
BGP table version is 20, local router ID is 172.16.7.7
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i  20.40.17.0/24    172.16.6.6               0    100      0 3356 1239 200 i
 *>                    40.60.27.2                            0 3356 1239 200 i
 * i  20.40.27.0/24    172.16.6.6               0    100      0 3356 1239 200 i
 *>                    40.60.27.2                            0 3356 1239 200 i
 * i  21.41.16.0/24    172.16.6.6               0    100      0 3356 1239 200 i
 *>                    40.60.27.2                            0 3356 1239 200 i
 * i  21.41.26.0/24    172.16.6.6               0    100      0 3356 1239 200 i
 *>                    40.60.27.2                            0 3356 1239 200 i
 * i  66.66.67.0/24    172.16.6.6               0    100      0 3356 1239 200 i
 *>                    40.60.27.2                            0 3356 1239 200 i
 * i  192.168.6.6/32   172.16.6.6               0    100      0 3356 1239 200 i
 *>                    40.60.27.2                            0 3356 1239 200 i
 * i  192.168.7.7/32   172.16.6.6               0    100      0 3356 1239 200 i
     Network          Next Hop            Metric LocPrf Weight Path
```

We need to prepare as-path regexp:

```
CEL7#conf t
CEL7(config)#ip as-path access-list 400 permit _200$
```

Next step is to apply it to BGP process in the form of route-map that will set traffic-index – attribute local to the router, that will enable counting of hits per matching entry:

```
CEL7(config)#route-map BGP-PA permit 10
CEL7(config-route-map)#match as-path 400
CEL7(config-route-map)#set traffic-index 30
CEL7(config-route-map)#route-map BGP-PA permit 20
CEL7(config-route-map)#router bgp 100
CEL7(config-router)#address-family ipv4
CEL7(config-router-af)#table-map BGP-PA
```

Last step is to activate BGP PA on interfaces – we're interested in traffic incoming from AS 200, so the direction is ingress to us (or input):

```
CEL7#conf t
CEL7(config)#int gi2
CEL7(config-if)#bgp accounting input
CEL7(config-if)#int gi3
CEL7(config-if)#bgp accounting input
CEL7(config-if)#int gi4
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CEL7(config-if)#bgp accounting input
CEL7(config-if)#int gi5
CEL7(config-if)#bgp accounting input
```

Let's clear BGP process to refresh the newly configures table-map, and check which prefixes fall into our new policy for traffic-index 30:

```
CEL7#clear ip bgp *
CEL7#
CEL7#
CEL7#sh ip cef 66.66.67.0/24 detail
66.66.67.0/24, epoch 2, flags [rib only nolabel, rib defined all labels]
  BGP: traffic_index 30
  recursive via 40.60.27.2
    attached to GigabitEthernet2
CEL7#sh ip cef 192.168.7.7/32 detail
192.168.7.7/32, epoch 2, flags [rib only nolabel, rib defined all labels]
  BGP: traffic_index 30
  recursive via 40.60.27.2
    attached to GigabitEthernet2
```

..and those not-matching regexp _200$, do not have traffic index:

```
CEL7#sh ip cef 33.56.0.0/16 detail
33.56.0.0/16, epoch 2, flags [rib only nolabel, rib defined all labels]
  recursive via 40.60.27.2
    attached to GigabitEthernet2
```

Now, let's generate some traffic from CER7 loopback traffic (192.168.7.7) towards AS100 and then check the stats:

```
CER7#traceroute 172.16.7.7 source loopback 0
Type escape sequence to abort.
Tracing the route to 172.16.7.7
VRF info: (vrf in name/id, vrf out name/id)
  1 20.40.27.2 3 msec 1 msec 1 msec
  2 12.39.23.3 [AS 1239] 4 msec 3 msec 4 msec
  3 22.22.33.1 6 msec 5 msec 5 msec
  4 33.56.13.1 [AS 3356] 7 msec 7 msec 7 msec
  5 41.61.16.6 [AS 100] 11 msec 8 msec 9 msec
  6 44.44.67.7 [AS 100] 9 msec *  10 msec
```

…and now check the BGP PA counters:

```
CEL7#show cef interface policy-statistics | i Gigab|BGP|30
GigabitEthernet1 is down (if_number 7)
GigabitEthernet2 is up (if_number 8)
  BGP based Policy accounting on input is enabled
    30               0               0
GigabitEthernet3 is up (if_number 9)
  BGP based Policy accounting on input is enabled
    30               3             168
GigabitEthernet4 is down (if_number 10)
  BGP based Policy accounting on input is enabled
    30               0               0
GigabitEthernet5 is up (if_number 11)
  BGP based Policy accounting on input is enabled
    30               0               0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.3: BGP policy accounting in IOS XR

We will execute the same BGP PA on IOS XR:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#as-path-set BGP-PA
RP/0/0/CPU0:SP1R2(config-as)#originates-from '200'
RP/0/0/CPU0:SP1R2(config-as)#end-set
RP/0/0/CPU0:SP1R2(config)#route-policy RP-BGP-PA
RP/0/0/CPU0:SP1R2(config-rpl)#if as-path in BGP-PA then
RP/0/0/CPU0:SP1R2(config-rpl-if)#set traffic-index 30
RP/0/0/CPU0:SP1R2(config-rpl-if)#else
RP/0/0/CPU0:SP1R2(config-rpl-else)#pass
RP/0/0/CPU0:SP1R2(config-rpl-else)#endif
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
```

And now the BGP configuration:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-af)#table-policy RP-BGP-PA
RP/0/0/CPU0:SP1R2(config-bgp-af)#commit
```

Lastly, interfaces enablement for input traffic:

```
RP/0/0/CPU0:SP1R2(config)#int gi0/0/0/0
RP/0/0/CPU0:SP1R2(config-if)#ipv4 bgp policy accounting input source-accounting
 ! note source-accounting or destination-accounting is non-relevant here, however
 ! can be used to further limit the accounting to specific role
RP/0/0/CPU0:SP1R2(config-if)#int gi0/0/0/2
RP/0/0/CPU0:SP1R2(config-if)#ipv4 bgp policy accounting input source-accounting
RP/0/0/CPU0:SP1R2(config-if)#int gi0/0/0/4
RP/0/0/CPU0:SP1R2(config-if)#ipv4 bgp policy accounting input source-accounting
RP/0/0/CPU0:SP1R2(config-if)#commit
```

Now, let's check if CEF properly reflects traffic index for prefixes originated from ASN 200:

```
RP/0/0/CPU0:SP1R2#sh cef ipv4 66.66.67.0/24
66.66.67.0/24, version 106, internal 0x1000001 0x0 (ptr 0xa1400e74) [1], 0x0 (0x0), 0x0
(0x0)
 Updated Jan  4 00:25:06.299
 local adjacency 33.56.23.3
 Prefix Len 24, traffic index 30, precedence n/a, priority 4
   via 33.56.3.3/32, 2 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa1400574 0x0]
    next hop 33.56.3.3/32 via 33.56.3.3/32

RP/0/0/CPU0:SP1R2#sh cef ipv4 192.168.7.7/32
192.168.7.7/32, version 116, internal 0x1000001 0x0 (ptr 0xa1400f74) [1], 0x0 (0x0), 0x0
(0x0)
 Updated Jan  4 00:25:06.300
 local adjacency 33.56.23.3
 Prefix Len 32, traffic index 30, precedence n/a, priority 4
   via 33.56.3.3/32, 2 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa1400574 0x0]
    next hop 33.56.3.3/32 via 33.56.3.3/32
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Just to check, other prefix that should be without traffic-index 30:

```
RP/0/0/CPU0:SP1R2#sh cef ipv4 12.39.0.0/16
12.39.0.0/16, version 88, internal 0x1000001 0x0 (ptr 0xa1400bf4) [1], 0x0 (0x0), 0x0
(0x0)
 Updated Jan  4 00:25:06.299
 local adjacency 33.56.23.3
 Prefix Len 16, traffic index 0, precedence n/a, priority 4
   via 33.56.3.3/32, 2 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa1400574 0x0]
    next hop 33.56.3.3/32 via 33.56.3.3/32
```

OK, so let's again generate some traffic SP1R2 bound from AS 200:

```
CER7#traceroute 33.56.2.2 source loopback 0
Type escape sequence to abort.
Tracing the route to 33.56.2.2
VRF info: (vrf in name/id, vrf out name/id)
  1 20.40.27.2 3 msec 2 msec 1 msec
  2 12.39.23.3 [AS 1239] 4 msec 4 msec 4 msec
  3 22.22.33.1 7 msec 6 msec 5 msec
  4 33.56.23.2 [AS 3356] 11 msec *  9 msec
```

And now let's check that the traffic is properly accounted for:

```
RP/0/0/CPU0:SP1R2#show cef ipv4 interface gigabitEthernet 0/0/0/0 bgp-policy-statistics
GigabitEthernet0/0/0/0 is UP
Input BGP policy accounting on src IP address enabled
  buckets      packets       bytes
  default           53        4500
  30                 3          84
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.4: CEF traffic accounting

You can ask CEF to keep statistics for per-prefix or per prefix-length traffic at router level. For IOS XE it's by default off:

```
CEL7#sh ip cef 66.66.67.0 detail
66.66.67.0/24, epoch 2, flags [rib only nolabel, rib defined all labels]
  recursive via 40.60.27.2
    attached to GigabitEthernet2
```

Once you turn it on (for IPv4 and IPv6 separately):

```
CEL7#conf t
CEL7(config)#ip cef accounting per-prefix non-recursive
CEL7(config)#ipv6 cef accounting per-prefix non-recursive
```

…and generate some traffic:

```
CEL7#ping 66.66.67.7 source lo0 size 1000 repeat 500
Type escape sequence to abort.
Sending 500, 1000-byte ICMP Echos to 66.66.67.7, timeout is 2 seconds:
Packet sent with a source address of 172.16.7.7
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[…]
!!!!!!!!!!
Success rate is 100 percent (500/500), round-trip min/avg/max = 9/10/16 ms
```

The traffic will be accounted for in CEF statistics:

```
CEL7#sh ip cef 66.66.67.0 detail
66.66.67.0/24, epoch 2, flags [rib only nolabel, rib defined all labels]
  500 packets, 500000 bytes switched to the prefix
  recursive via 40.60.27.2
    attached to GigabitEthernet2
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4: Monitoring network conditions and acting on results

Task 4.1: IP SLA in IOS-XE

IP SLA is built in tool to monitor various networking environment characteristics using probes sent from the monitoring mode (and eventually looped or responded to by receiving hosts).

To define basic IP SLA probe from CEL6 to CEL7 using ICMP echo packets and loopback 0 addresses, and then schedule it to run forever, you need to configure following options:

```
CEL6(config)#ip sla 100
CEL6(config-ip-sla)#icmp-echo 172.16.7.7 source-ip 172.16.6.6
CEL6(config-ip-sla-echo)#frequency 5
CEL6(config-ip-sla-echo)#exit
CEL6(config)#ip sla schedule 100 start-time now life forever
```

To check its operation:

```
CEL6#sh ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 100
        Latest RTT: 2 milliseconds
Latest operation start time: 10:06:05 UTC Fri Jan 01 2017
Latest operation return code: OK
Number of successes: 5
Number of failures: 0
Operation time to live: Forever
```

Now let's shut down CEL7 lo0 interface:

```
CEL7(config)#int lo0
CEL7(config-if)#shut
```

We see that the probe now fails:

```
CEL6#sh ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 100
        Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: 10:06:45 UTC Fri Jan 01 2017
Latest operation return code: Timeout
Number of successes: 11
Number of failures: 2
Operation time to live: Forever
```

Let's bring the CEL7 lo0 interface back up:

```
CEL7(config)#int lo0
CEL7(config-if)#no shut
```

Now, we'll define more complex probe using tracking functionality. We'll not only check if the CEL7 lo0 is available, but also if interface between CEL6 and CEL7 is available.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

If either of those conditions fails, we'll declare tracking object down:

```
CEL6(config)#track 1 ip sla 100
CEL6(config)#track 2 interface gigabitEthernet 3 line-protocol
CEL6(config)#track 10 list boolean and
CEL6(config-track)#object 1
CEL6(config-track)#object 2
CEL6(config-track)#^Z
```

Now, check the tracking status:

```
CEL6#sh track
Track 1
  IP SLA 100 state
  State is Up
    1 change, last change 00:01:35
  Latest operation return code: OK
  Latest RTT (millisecs) 1
  Tracked by:
    Track List 10
Track 2
  Interface GigabitEthernet3 line-protocol
  Line protocol is Up
    1 change, last change 00:01:29
  Tracked by:
    Track List 10
Track 10
  List boolean and
  Boolean AND is Up
    2 changes, last change 00:00:13
    object 1 Up
    object 2 Up
```

We can see that the master object 10 is up as both of its components (track 1 and track 2) are up.

Let's shut down the Lo0 on CEL7 again to test the tracking object 10:

```
CEL7(config)#int lo0
CEL7(config-if)#shut
```

Now let's monitor CEL6:

```
000142: *Jan 01 10:14:30.331 UTC: %TRACK-6-STATE: 1 ip sla 100 state Up -> Down
000143: *Jan 01 10:14:31.310 UTC: %TRACK-6-STATE: 10 list boolean and Up -> Down
CEL6#sh track 10
Track 10
  List boolean and
  Boolean AND is Up
    3 changes, last change 00:00:18
    object 1 Down
    object 2 Up
```

The tracking object can now be used for example in tracking route entries, or within EEM as we'll see in next exercise.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

For now, let’s bring the lo0 on CEL7 back again and observe the situation:

```
000154: *Jan 01 10:17:35.336 UTC: %TRACK-6-STATE: 1 ip sla 100 state Down -> Up
000155: *Jan 01 10:17:35.340 UTC: %TRACK-6-STATE: 10 list boolean and Down -> Up
CEL6#sh track 10
Track 10
  List boolean and
  Boolean AND is Up
    4 changes, last change 00:00:01
    object 1 Up
    object 2 Up
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.2: IP SLA in IOS XR

IP SLA for IOS XR has less capabilities, but basic checks are functional and supported. For example, to schedule basic ICMP echo test from R2 in AS1239 to R2 in AS3356 we'll define:

```
RP/0/0/CPU0:SP2R2(config)#ipsla
RP/0/0/CPU0:SP2R2(config-ipsla)#operation 100
RP/0/0/CPU0:SP2R2(config-ipsla-op)#type icmp echo
RP/0/0/CPU0:SP2R2(config-ipsla-icmp-echo)#source address 12.39.2.2
RP/0/0/CPU0:SP2R2(config-ipsla-icmp-echo)#destination address 33.56.1.1
RP/0/0/CPU0:SP2R2(config-ipsla-icmp-echo)#frequency 5
RP/0/0/CPU0:SP2R2(config-ipsla-icmp-echo)#exit
RP/0/0/CPU0:SP2R2(config-ipsla-op)#schedule operation 100
RP/0/0/CPU0:SP2R2(config-ipsla-sched)#start-time now
RP/0/0/CPU0:SP2R2(config-ipsla-sched)#life forever
RP/0/0/CPU0:SP2R2(config-ipsla-sched)#commit
```

And to check status of probe:

```
RP/0/0/CPU0:SP2R2#show ipsla statistics
Entry number: 100
    Modification time: 20:18:14.172 UTC Fri Jan 01 2017
    Start time       : 20:18:14.192 UTC Fri Jan 01 2017
    Number of operations attempted: 10
    Number of operations skipped  : 0
    Current seconds left in Life  : Forever
    Operational state of entry    : Active
    Operational frequency(seconds): 5
    Connection loss occurred      : FALSE
    Timeout occurred              : FALSE
    Latest RTT (milliseconds)     : 1
    Latest operation start time   : 20:18:59.428 UTC Fri Jan 01 2017
    Next operation start time     : 20:19:04.428 UTC Fri Jan 01 2017
    Latest operation return code  : OK
    RTT Values:
      RTTAvg  : 1          RTTMin: 1          RTTMax : 1
      NumOfRTT: 1          RTTSum: 1          RTTSum2: 1
```

IOS XR also has the tracking feature, but as with IP SLA – it's much simplified version of the feature compared to IOS-XE. For example, to track status of link:

```
RP/0/0/CPU0:SP2R2(config)#track GI00 type line-protocol state interface gig 0/0/0/0
```

And to check the status:

```
RP/0/0/CPU0:SP2R2#sh track

Track GI00
        Interface GigabitEthernet0/0/0/0 line-protocol
        Line protocol is UP
        1 change, last change 20:22:46 UTC Fri Jan 01 2017
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.3: EEM

Embedded Event Manager is simple, but quite powerful tool to automate different tasks and enhance operation of device. You can fire EEM on specific events, or make it react to some operator actions – or even use it as periodically run application that will check some properties.

Task 4.3.1: Basic EEM script

Now we can check if there's any traffic coming over CEL6:

Let's define basic EEM script, that will run every 30 seconds and send information to syslog buffer:

```
CEL6(config)#event manager applet W30
CEL6(config-applet)#event timer watchdog time 30
CEL6(config-applet)#action 10 syslog msg "Watchdog executed EEM!"
CEL6(config-applet)#^Z
```

Now, let's observe buffer:

```
000157: *Jan 01 13:32:18.413 UTC: %HA_EM-6-LOG: W30: Watchdog executed EEM!
000158: *Jan 01 13:32:48.413 UTC: %HA_EM-6-LOG: W30: Watchdog executed EEM!
000159: *Jan 01 13:33:18.413 UTC: %HA_EM-6-LOG: W30: Watchdog executed EEM!
000160: *Jan 01 13:33:48.413 UTC: %HA_EM-6-LOG: W30: Watchdog executed EEM!
000161: *Jan 01 13:34:18.413 UTC: %HA_EM-6-LOG: W30: Watchdog executed EEM!
```

Task 4.3.2: More advanced example – network interfaces

The more complex scenario we'll go over is script that will shut down specific interface once some test fail – we'll use the IP SLA probes and track objects defined previously to trigger EEM run:

```
CEL6(config)#no event manager applet W30
CEL6(config)#event manager applet CHECK-INTF
CEL6(config-applet)#event track 10 state down
CEL6(config-applet)#action 10 cli command "enable"
CEL6(config-applet)#action 15 cli command "conf t"
CEL6(config-applet)#action 20 cli command "int lo0"
CEL6(config-applet)#action 25 cli command "shut"
CEL6(config-applet)#action 30 syslog msg "Lo0 shut down as Track 10 is down!"
CEL6(config-applet)#^Z
```

Let's check status of track 10:

```
CEL6#sh track 10
Track 10
  List boolean and
  Boolean AND is Up
    4 changes, last change 03:19:25
    object 1 Up
    object 2 Up
  Tracked by:
    EEM 1666508456 ! EEM script registered itself as tracking this object
```

Now, let's make track 10 to fail – by shutting down Lo0 on CEL7:

```
CEL7(config)#int lo0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
CEL7(config-if)#shut
CEL7(config-if)#^Z
```

The EEM will trigger immediately:

```
000168: *Jan 01 13:37:41.070 UTC: %HA_EM-6-LOG: CHECK-INTF: Lo0 shut down as Track 10 is
down!
```

…and interface lo0 is down:

```
CEL6#sh int lo0
Loopback0 is administratively down, line protocol is down
```

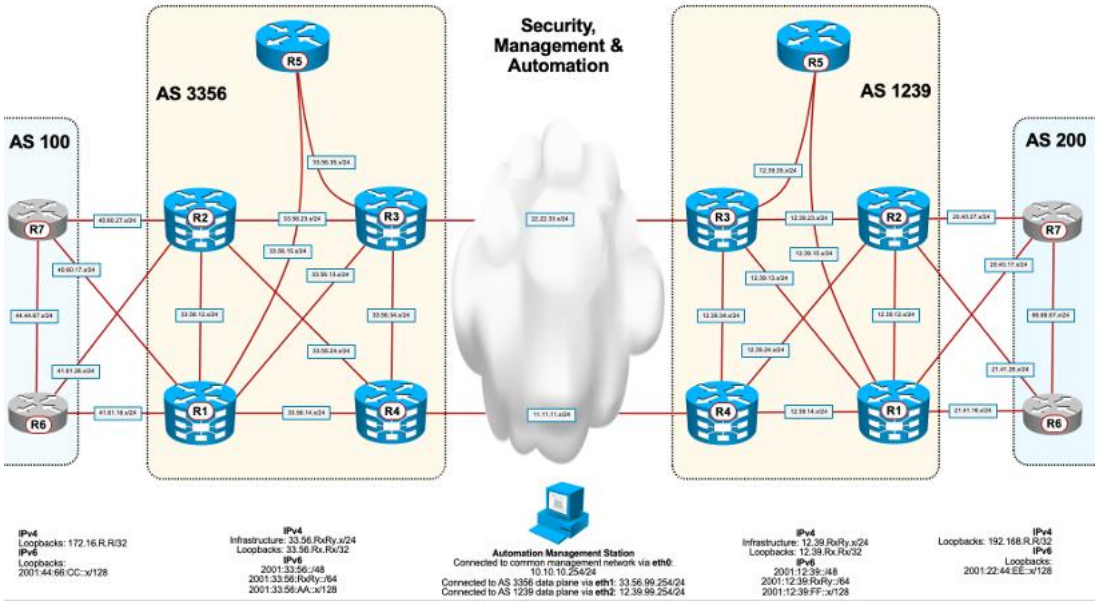
Task 4.3.3: Building script to bring Lo0 back up again once Track 10 is up

Now, as matter of exercise, define another EEM script that will react to tracking 10 being brought up and bring back Lo0 to up state.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 18: System and network security

For system and network we'll use following topology:



This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch18.virl1.virl`
- **For VIRL²/CML²:** `sp.bootcamp.ch18.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch18.unl`

Task 1: Securing control plane

Task 1.1: CoPP

CoPP is feature that can protect router RP from attack. It's stateless, but given configuration options to limit traffic both in throughput as well as in pps it's useful to define acceptable ranges for specific traffic types, once you know baseline.

Let's define ACLs that will control traffic types directed at or hitting the control plane:

```
CEL6(config)#ip access-list extended ACL-COPP-ROUTING
CEL6(config-ext-nacl)#permit tcp any any eq 179
CEL6(config-ext-nacl)#permit tcp any eq 179 any
CEL6(config-ext-nacl)#permit ospf any any
CEL6(config-ext-nacl)#permit eigrp any any
CEL6(config)#ip access-list extended ACL-COPP-DIAG
CEL6(config-ext-nacl)#permit icmp any any
CEL6(config)#ip access-list extended ACL-COPP-OTHER
CEL6(config-ext-nacl)#permit ip any any
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Now, let's define class-maps, as CoPP uses MQC to apply traffic policies. Each class-map will match specific ACL:

```
CEL6(config)#class-map CM-COPP-ROUTING
CEL6(config-cmap)#match access-group name ACL-COPP-ROUTING
CEL6(config-cmap)#class-map CM-COPP-DIAG
CEL6(config-cmap)#match access-group name ACL-COPP-DIAG
CEL6(config-cmap)#class-map CM-COPP-OTHER
CEL6(config-cmap)#match access-group name ACL-COPP-OTHER
```

At last, define the policy-map for the defined class-maps:

```
CEL6(config)#policy-map PM-COPP
CEL6(config-pmap)#class CM-COPP-ROUTING
CEL6(config-pmap-c)#police 32k
CEL6(config-pmap-c-police)#class CM-COPP-DIAG
CEL6(config-pmap-c)#police 8000
CEL6(config-pmap-c-police)#class CM-COPP-OTHER
CEL6(config-pmap-c)#police 8000
CEL6(config-pmap-c-police)#exit
CEL6(config-pmap-c)#^Z
```

Now, let's apply the PM-COPP as input policy for control plane of the router:

```
CEL6#conf t
CEL6(config)#control-plane
CEL6(config-cp)#service-policy input PM-COPP
CEL6(config-cp)#^Z
```

Once you attach the service policy to control plane, using normal MQC commands you can check the effects on the router:

```
CEL6#sh policy-map control-plane
 Control Plane

  Service-policy input: PM-COPP

    Class-map: CM-COPP-ROUTING (match-all)
      41 packets, 3686 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name ACL-COPP-ROUTING
      police:
          cir 32000 bps, bc 1500 bytes
        conformed 41 packets, 3686 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceeded 0000 bps

    Class-map: CM-COPP-DIAG (match-all)
      75 packets, 6012 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name ACL-COPP-DIAG
      police:
          cir 8000 bps, bc 1500 bytes
        conformed 75 packets, 6012 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceeded 0000 bps

    Class-map: CM-COPP-OTHER (match-all)
      64 packets, 4734 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name ACL-COPP-OTHER
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      police:
          cir 8000 bps, bc 1500 bytes
        conformed 64 packets, 4734 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceeded 0000 bps

    Class-map: class-default (match-any)
      170 packets, 5478 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

We're matching some normal traffic, way below limits defined for policies. Now, let's generate a lot of ICMP traffic going towards CEL6 from CEL7:

```
CEL7#ping 172.16.6.6 source lo0 size 1500 repeat 500
Type escape sequence to abort.
Sending 500, 1500-byte ICMP Echos to 172.16.6.6, timeout is 2 seconds:
Packet sent with a source address of 172.16.7.7
!.!.!.!.!.!.!.!
```

We can see that each other packet overflows CoPP on CEL6 and causes router to drop it. We can verify this on CEL6 side:

```
CEL6#sh policy-map control-plane
 Control Plane

  Service-policy input: PM-COPP

[…]

    Class-map: CM-COPP-DIAG (match-all)
      173 packets, 79176 bytes
      5 minute offered rate 4000 bps, drop rate 4000 bps
      Match: access-group name ACL-COPP-DIAG
      police:
          cir 8000 bps, bc 1500 bytes
        conformed 150 packets, 44676 bytes; actions:
          transmit
        exceeded 23 packets, 34500 bytes; actions:
          drop
        conformed 4000 bps, exceeded 4000 bps

[…]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.2: LPTS

LTPS on IOS XR is way more advanced version of CoPP – it's stateful, it's able to accommodate configuration of node automatically, and it can also be tweaked. You can verify current LPTS bindings using:

```
RP/0/0/CPU0:SP1R1#show lpts bindings brief

@ - Indirect binding; Sc - Scope

 Location   Clnt Sc L3   L4     VRF-ID    Interface    Local-Address,Port Remote-
Address,Port
 ---------- ---- -- ---- ------ --------- ------------ --------------------------------
-----
 0/0/CPU0   IPV4 LO IPV4 ICMP   *         any          any,ECHO any
 0/0/CPU0   IPV4 LO IPV4 ICMP   *         any          any,TSTAMP any
 0/0/CPU0   IPV4 LO IPV4 ICMP   *         any          any,MASKREQ any
 0/0/CPU0   IPV6 LO IPV6 ICMP6  *         any          any,ECHOREQ any
 0/0/CPU0   IPV6 LO IPV6 ICMP6  *         any          any,NDRTRSLCT any
 0/0/CPU0   IPV6 LO IPV6 ICMP6  *         any          any,NDRTRADV any
 0/0/CPU0   IPV6 LO IPV6 ICMP6  *         any          any,NDNBRSLCT any
 0/0/CPU0   IPV6 LO IPV6 ICMP6  *         any          any,NDNBRADV any
 0/0/CPU0   IPV6 LO IPV6 ICMP6  *         any          any,NDREDIRECT any
 0/0/CPU0   BFD  LO IPV4 UDP    *         any          any any
```

The command can also be extended to match specific protocols and interfaces, like this (for ICMP – protocol number 1):

```
RP/0/0/CPU0:SP1R1#show lpts bindings brief location 0/0/CPU0 l4-protocol 1

@ - Indirect binding; Sc - Scope

 Clnt Sc L3   L4     VRF-ID              Interface    Local-Address,Port Remote-
Address,Port
 ---- -- ---- ------ ------------------- ------------ --------------------------------
-----
 IPV4 LO IPV4 ICMP   *                   any          any,ECHO any
 IPV4 LO IPV4 ICMP   *                   any          any,TSTAMP any
 IPV4 LO IPV4 ICMP   *                   any          any,MASKREQ any
```

…or TCP (protocol number 6):

```
RP/0/0/CPU0:SP1R1#show lpts bindings brief location 0/0/CPU0 l4-protocol 6

@ - Indirect binding; Sc - Scope

 Clnt Sc L3   L4     VRF-ID              Interface    Local-Address,Port Remote-
Address,Port
 ---- -- ---- ------ ------------------- ------------ --------------------------------
-----
 TCP  LR IPV6 TCP    default             Gi0/0/0/0    any,22 any
 TCP  LR IPV4 TCP    default             Gi0/0/0/0    any,22 any
 TCP  LR IPV4 TCP    default             any          33.56.1.1,56265 33.56.5.5,179
 TCP  LR IPV4 TCP    default             any          41.61.16.1,29969 41.61.16.6,179
 TCP  LR IPV4 TCP    default             any          33.56.1.1,646 33.56.5.5,54329
 TCP  LR IPV6 TCP    default             any          2001:41:61:16::1,35041
2001:41:61:16::6,179
 TCP  LR IPV4 TCP    default             any          33.56.1.1,646 33.56.4.4,23250
 TCP  LR IPV4 TCP    default             any          any,646 33.56.2.2
[…]
 TCP  LR IPV4 TCP    default             any          33.56.1.1,646 33.56.4.4,58632
 TCP  LR IPV6 TCP    Mgmt-intf           Mg0/0/CPU0/0 any,22 any
 TCP  LR IPV4 TCP    Mgmt-intf           Mg0/0/CPU0/0 any,22 any
 TCP  LR IPV4 TCP    Mgmt-intf           Mg0/0/CPU0/0 any,23 any
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

In XRv there's no option to change the punt parameters, so apart from verification and/or troubleshooting, we can't go deeper or reconfigure anything.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3: BGP security

Routing protocols can be abused – specifically at the edges of your domain, but not only there. As BGP is used to exchange external routes with world and often because of flexible policy options used as PE-CE protocol, securing it down is one of the most universal and commonly done things when thinking about security of your routing domain.

Task 1.3.1: BGP maximum prefix feature on IOS XE

One of the things impacting potential stability of the platform is crossing the platform capacity to hold routing data – FIB size. Main compound is usually BGP, holding both global, internet prefixes (700k+ for IPv4 and 16k+ for IPv6) as well as VPN prefixes.

Lets configure limit for external neighbor on the number of prefixes that can be advertised for CER6 router. Right now, CER6 has 11 prefixes (this may vary in your specific config, depending on where you started with the config, please adjust to observe similar results):

```
CER6#sh ip bgp summary
BGP router identifier 192.168.6.6, local AS number 200
BGP table version is 19, main routing table version 19
18 network entries using 4464 bytes of memory
30 path entries using 4080 bytes of memory
8/5 BGP path/bestpath attribute entries using 2240 bytes of memory
3 BGP AS-PATH entries using 104 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 10888 total bytes of memory
BGP activity 34/0 prefixes, 56/0 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
21.41.16.1      4         1239      17      18       19    0    0 00:12:31        11
192.168.7.7     4          200      19      19       19    0    0 00:13:19        15
```

So, with 11 prefixes lets limit our config to 10 and generate warning only – without restarting the session:

```
CER6(config)#router bgp 200
CER6(config-router)#address-family ipv4 unicast
CER6(config-router-af)#neighbor 21.41.16.1 maximum-prefix 10 ?
  <1-100>       Threshold value (%) at which to generate a warning msg
  restart       Restart bgp connection after limit is exceeded
  warning-only  Only give warning message when limit is exceeded
  <cr>
CER6(config-router-af)#neighbor 21.41.16.1 maximum-prefix 10 75 warning-only
```

Depending on the number of prefixes you currently have, you'l

```
<timestamp>: %BGP-3-MAXPFXEXCEED: Number of prefixes received from 21.41.16.1 (afi 0):
11 exceeds limit 10
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3.2: BGP VRF route limit on IOS XE

Other way to limit number of routes is to limit them on per-AF, per-VRF level. This can be achieved in a way similar to the previous configuration, but at VRF configuration level:

```
CER6#conf t
CER6(config)#vrf definition ABC1
CER6(config-vrf)#address-family ipv4 unicast
CER6(config-vrf-af)#maximum routes 500 75
CER6(config-vrf)#address-family ipv6 unicast
CER6(config-vrf-af)#maximum routes 500 75
```

This can be then verified with:

```
CER6#sh vrf detail ABC1
VRF ABC1 (VRF Id = 2); default RD <not set>; default VPNID <not set>
  New CLI format, supports multiple address-families
  Flags: 0x1808
  No interfaces
Address family ipv4 unicast (Table ID = 0x2):
  Flags: 0x0
  No Export VPN route-target communities
  No Import VPN route-target communities
  No import route-map
  No global export route-map
  No export route-map
  Route warning limit 500, current count 392 ! note with clean VRF count will be 0
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
Address family ipv6 unicast (Table ID = 0x1E000002):
  Flags: 0x0
  No Export VPN route-target communities
  No Import VPN route-target communities
  No import route-map
  No global export route-map
  No export route-map
  Route warning limit 500, current count 211 ! note with clean VRF count will be 0
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
Address family ipv4 multicast not active
Address family ipv6 multicast not active
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3.3: BGP maximum prefix feature on IOS XR

Lets test the same configuration for IOS XR – this time on SP1R4, receiving 8 prefixes from AS 1239:

```
RP/0/0/CPU0:SP1R4#sh bgp ipv4 unicast summary
[…]

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
11.11.11.254      0  1239       9       8       43    0    0 00:02:43          8
33.56.5.5         0  3356      11       8       43    0    0 00:02:32         18
```

The configuration looks very similar:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#router bgp 3356
RP/0/0/CPU0:SP1R4(config-bgp)#neighbor 11.11.11.254
RP/0/0/CPU0:SP1R4(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R4(config-bgp-nbr-af)#maximum-prefix 5 ?
  <1-100>              Threshold value (%) at which to generate a warning msg
  discard-extra-paths  Discard extra paths when limit is exceeded
  restart              Restart time interval
  warning-only         Only give warning message when limit is exceeded
  <cr>
RP/0/0/CPU0:SP1R4(config-bgp-nbr-af)#maximum-prefix 5 75 warning-only
RP/0/0/CPU0:SP1R4(config-bgp-nbr-af)#commit
RP/0/0/CPU0:SP1R4(config-bgp-nbr-af)#
```

And again, the warning should be printed right away on the console:

```
RP/0/0/CPU0: bgp[1052]: %ROUTING-BGP-4-MAXPFXEXCEED : No. of IPv4 Unicast prefixes
received from 11.11.11.254: 8 exceed limit 5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3.4: BGP VRF route limit on IOS XR

Other way to limit number of routes is to limit them on per-AF, per-VRF level. This can be achieved in a way similar to the previous configuration, but at VRF configuration level:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#vrf ABC1
RP/0/0/CPU0:SP1R4(config-vrf)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R4(config-vrf-af)#maximum prefix 500 ?
  <1-100>  mid-thresh (% of max)
  <cr>
RP/0/0/CPU0:SP1R4(config-vrf-af)#maximum prefix 500 75
RP/0/0/CPU0:SP1R4(config-vrf-af)#exit
RP/0/0/CPU0:SP1R4(config-vrf)#address-family ipv6 unicast
RP/0/0/CPU0:SP1R4(config-vrf-af)#maximum prefix 500 75
RP/0/0/CPU0:SP1R4(config-vrf-af)#exit
RP/0/0/CPU0:SP1R4(config-vrf)#commit
```

This can be then verified with:

```
RP/0/0/CPU0:SP1R4#show rib tables

Codes: N - Prefix Limit Notified, F - Forward Referenced
       D - Table Deleted, C - Table Reached Convergence

VRF/Table              SAFI  Table ID     PrfxLmt   PrfxCnt TblVersion  N F D C
default/default        uni   0xe0000000   5000000        38         73  N N N Y
**nVSatellite/default  uni   0xe0000010   5000000         2          4  N N N Y
ABC1/default           uni   0xe0000011       500         0          0  N N N Y
default/default        multi 0xe0100000   5000000         0          0  N N N Y
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3.5: BGP attribute limits

BGP can also be protected at the attribute level. Usual problems start when misconfigured system in the internet sends very long AS_PATH attribute, or attaches to many communities. With IOS and IOS-XE this can be controlled with global commands:

```
CEL7(config)#router bgp 100
CEL7(config-router)#bgp maxas-limit ?
  <1-254>  Number of ASes in the AS-PATH attribute(max 254)
CEL7(config-router)#bgp maxas-limit 16

CEL7(config-router)#bgp maxcommunity-limit ?
  <1-1018>  Number of communities in the COMMUNITY attribute
CEL7(config-router)#bgp maxcommunity-limit 16

CEL7(config-router)#bgp maxextcommunity-limit ?
  <1-509>  Number of extended communities in the EXTENDED COMMUNITY attribute
CEL7(config-router)#bgp maxextcommunity-limit 16
```

For IOS XR, the MAX AS limit can be enforced using RPL:

```
RP/0/0/CPU0:SP1R2(config-rpl)#if as-path length ge 16 then
RP/0/0/CPU0:SP1R2(config-rpl-if)#drop
RP/0/0/CPU0:SP1R2(config-rpl-if)#endif
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Securing management plane

Task 2.1: MPP

Task 2.1.1: MPP on IOS-XE

Management Plane Protection is capability of Control Plane Protection within IOS and IOS-XE. It allows globally defining interfaces over which incoming sessions (no matter to which router IPs) can be permitted. For example, let's define Gi3 as the only interface over which incoming SSH requests can come in on CEL6:

```
CEL6(config)#control-plane host
CEL6(config-cp-host)#management-interface gigabitEthernet 3 allow ?
  beep     Beep Protocol
  ftp      File Transfer Protocol
  http     HTTP Protocol
  https    HTTPS Protocol
  snmp     Simple Network Management Protocol
  ssh      Secure Shell Protocol
  telnet   Telnet Protocol
  tftp     Trivial File Transfer Protocol
  tl1      Transaction Language Session Protocol
  <cr>
CEL6(config-cp-host)#management-interface gigabitEthernet 3 allow ssh
```

Note, that if incoming sessions to router IP will arrive over other interfaces, they'll be blocked.

Task 2.1.2: MPP on IOS XR

Similar capability is available in IOS XR – with the distinction between in-band and out-of-band options:

```
RP/0/0/CPU0:SP1R1(config)#control-plane management-plane
RP/0/0/CPU0:SP1R1(config-mpp)#inband interface gigabitEthernet 0/0/0/0
RP/0/0/CPU0:SP1R1(config-mpp-inband-if)#allow ?
  HTTP     HTTP(S)
  NETCONF  NETCONF version 1.1 protocol
  SNMP     SNMP (all versions)
  SSH      Secure Shell (v1 & v2)
  TFTP     TFTP
  Telnet   Telnet
  XML      XML
  all      All Protocols
RP/0/0/CPU0:SP1R1(config-mpp-inband-if)#allow SSH
RP/0/0/CPU0:SP1R1(config-mpp-inband-if)#commit
```

You can verify the configuration:

```
RP/0/0/CPU0:SP1R1#show mgmt-plane

Management Plane Protection

inband interfaces
-----------------------
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
interface - GigabitEthernet0/0/0/0
        ssh configured -
                All peers allowed

interface - all
        all configured -
                All peers allowed
```

While inband configuration touches interfaces that will process data traffic, out-of-band interfaces are dedicated to processing management traffic. For that, the best practice is to use separate VRF instance, with separate routing and forwarding table just in case something happens to global connectivity. Let's do that with default VRF that physical Management interfaces belong to (Mgmt-intf). First, we'll delete configuration that makes all interfaces belong to inband (interfaces can't be both inband and out-of-band):

```
RP/0/0/CPU0:SP1R1(config)#control-plane
RP/0/0/CPU0:SP1R1(config-ctrl)#management-plane
RP/0/0/CPU0:SP1R1(config-mpp)#inband
RP/0/0/CPU0:SP1R1(config-mpp-inband)#no interface all
RP/0/0/CPU0:SP1R1(config-mpp-inband)#commit
RP/0/0/CPU0:SP1R1(config-mpp-inband)#
```

Now, let's add the Management interface to out-of-band category:

```
RP/0/0/CPU0:SP1R1(config)#control-plane
RP/0/0/CPU0:SP1R1(config-ctrl)#management-plane
RP/0/0/CPU0:SP1R1(config-mpp)#out-of-band
RP/0/0/CPU0:SP1R1(config-mpp-outband)#interface mgmtEth 0/0/CPU0/0 allow ssh
RP/0/0/CPU0:SP1R1(config-mpp-outband)#commit
RP/0/0/CPU0:SP1R1(config-mpp-outband)#
```

And now let's verify it:

```
RP/0/0/CPU0:SP1R1#show mgmt-plane out-of-band interface all

Management Plane Protection

outband interfaces
-----------------------

interface - MgmtEth0/0/CPU0/0
        ssh configured -
                All peers allowed
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.2: Telnet and SSH

Task 2.2.1: Telnet and SSH on IOS-XE

Most basic way of enabling or disabling access to the router using Telnet or SSH is by configuring VTYs:

```
CEL6(config)#line vty 0 ?
  <1-98>  Last Line number
  <cr>
CEL6(config)#line vty 0 98
CEL6(config-line)#transport input ?
[…]
  all      All protocols
[…]
  none     No protocols
[…]
  ssh      TCP/IP SSH protocol
  telnet   TCP/IP Telnet protocol
CEL6(config-line)#transport input telnet ssh
```

For SSH to work on IOS/IOS-XE you need to generate crypto keys – and for that, router needs hostname and ip domain defined:

```
CEL6(config)#crypto key generate rsa
The name for the keys will be: CEL6.netdesign.zone
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 0 seconds)
```

By default, IOS uses SSH v1 (v1.99), and to enable SSH v2 you need to change the default:

```
CEL6(config)#ip ssh version 2
CEL6(config)#do sh ip ssh
SSH Enabled - version 2.0
Authentication methods:password
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa
Hostkey Algorithms:x509v3-ssh-rsa,ssh-rsa
Encryption Algorithms:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-
cbc,aes256-cbc
MAC Algorithms:hmac-sha1,hmac-sha1-96
KEX Algorithms:diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 2048 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded): TP-self-signed-391156546
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDXZ/WBR2Vlw8XI9rm6nFoTB/4AcOqjz0E6fQCaOihd
+72xigvBUP1Xced+ZP4653GrzLX30ciz/dj8lE09+R7qSrEDLu/T1YemyiytB19FzDKslV2qa0JZEwxX
8A3wbWwYJxILM/XXxE2YGx7Fs50RcZs3QnPFiy5YLRDBw3Jm++aKCu/rIGEBtcTMjRc+fPhb/Jzr3r/A
z5z5t1S/oMuDFfk4xt4/YJj081VMUKzEmEXn1nf01PniUSvPdXbS2hRjTN6hLFm+FK0Jkd+tvHZXTUB9
jTVgD/HYxTHFwRdHc46grpvbuEtGBgYQGuLO/qvmj31kJcJLfvYACjLcGOPx
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.2.2: Telnet and SSH on IOS XR

For IOS XR, most of the features are running their own servers (processes) within OS. For Telnet, this means you need to define number of telnet servers (and protocol over which they will be accessible) to enable Telnet on router:

```
RP/0/0/CPU0:SP1R1(config)#telnet ipv4 server max-servers 10
RP/0/0/CPU0:SP1R1(config)#telnet ipv6 server max-servers 10
RP/0/0/CPU0:SP1R1(config)#commit
```

For SSH to start, we need as with IOS/IOS-XE – hostname, domain name and generating keys for SSH to use (you're doing that in global mode, not in configuration mode!):

```
RP/0/0/CPU0:SP1R1#crypto key generate rsa
The name for the keys will be: the_default
  Choose the size of the key modulus in the range of 512 to 4096 for your General
Purpose Keypair. Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [2048]: 1024
Generating RSA keys ...
Done w/ crypto generate keypair
[OK]
```

Now, we need to enable SSH server:

```
RP/0/0/CPU0:SP1R1#conf t
RP/0/0/CPU0:SP1R1(config)#ssh server v2
```

Note, that by default, IOS XR version of SSH implementation has extensive amount of hardening options – including ACLs and session/rate-limiting:

```
RP/0/0/CPU0:SP1R1(config)#ssh server ?
  dscp           Cisco ssh server DSCP
  ipv4           IPv4 access list for ssh server
  ipv6           IPv6 access list for ssh server
  logging        Enable ssh server logging
  netconf        start ssh service for netconf subsystem
  rate-limit     Cisco sshd rate-limit of service requests
  session-limit  Cisco sshd session-limit of service requests
  v2             Cisco sshd force protocol version 2 only
  vrf            Cisco sshd VRF name
  <cr>
RP/0/0/CPU0:SP1R1(config)#ssh server rate-limit ?
  <1-600>  Limit in term of number of request per minute (default 60)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3: Securing forwarding plane

Task 3.1: uRPF

Task 3.1.1: uRPF on IOS XR

To configure IOS XR for uRPF you need to enable it per interface:

```
RP/0/0/CPU0:SP1R1(config)#interface gigabitEthernet 0/0/0/0
RP/0/0/CPU0:SP1R1(config-if)#ipv4 verify unicast source reachable-via rx
RP/0/0/CPU0:SP1R1(config-if)#commit
```

We'll also add static route to make sure we will be able to test uRPF failure:

```
RP/0/0/CPU0:SP1R1(config)#router static
RP/0/0/CPU0:SP1R1(config-static)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R1(config-static-afi)#33.56.4.4/32 33.56.2.2
RP/0/0/CPU0:SP1R1(config-static-afi)#commit
```

We can verify the configuration:

```
RP/0/0/CPU0:SP1R1#sh ipv4 interface gigabitEthernet 0/0/0/0
GigabitEthernet0/0/0/0 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 33.56.14.1/24
  MTU is 1514 (1500 is available to IP)
[..]
  Inbound  common access list is not set, access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
IP unicast RPF check is enabled
RPF mode strict
```

On SP1R4 pings to SP1R1 loopback will now fail because of static route:

```
RP/0/0/CPU0:SP1R4#ping 33.56.1.1 source 33.56.4.4 repeat 10 size 200
Type escape sequence to abort.
Sending 10, 200-byte ICMP Echos to 33.56.1.1, timeout is 2 seconds:
..........
Success rate is 0 percent (0/10)
```

While on SP1R1 we can observe uRPF failures:

```
RP/0/0/CPU0:SP1R1#show cef ipv4 interface gigabitEthernet 0/0/0/0 rpf-statistics
Unicast RPF drops 77
```

You can also check them globally for the node:

```
RP/0/0/CPU0:SP1R1#show cef ipv4 drops
CEF Drop Statistics
Node: 0/0/CPU0
[…]
  RPF drops             packets :               79
[…]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.1.2: uRPF on IOS/IOS-XE

uRPF configuration on IOS and IOS-XE is very similar to that of IOS XR.

Let's use CEL6 and CEL7 to check the config, this time with IPv6. Also, we'll not use strict but loose checking, defining the static route this time to Null0 interface.

On CEL6 we point CEL7 IPv6 loopback address to Null0:

```
CEL6(config)#ipv6 route 2001:44:66:CC::7/128 null 0
```

Now, let's enable IPv6 uRPF in loose mode on Gi3 interface, connecting CEL6 and CEL7:

```
CEL6#conf t
CEL6(config)#int gi3
CEL6(config-if)#ipv6 verify unicast source reachable-via any
```

Now, any IPv6 traffic from CEL7 lo0 interface towards CEL6 should be dropped by uRPF as not passing the check (Null0 can't be valid source for any packet!):

```
CEL7#ping 2001:44:66:CC::6 source lo0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:44:66:CC::6, timeout is 2 seconds:
Packet sent with a source address of 2001:44:66:CC::7
.....
```

Check on the CEL6 Gi3 interface:

```
CEL6#sh ipv6 interface gigabitEthernet 3
GigabitEthernet3 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::6
  No Virtual link-local address(es):
  Description: to CEL7
  Global unicast address(es):
    2001:44:44:67::6, subnet is 2001:44:44:67::/64
[…]
  ICMP unreachables are sent
  Input features: Verify Unicast Reverse-Path
 IPv6 verify source reachable-via any
   0 verification drop(s) (process), 5 (CEF)
   0 suppressed verification drop(s) (process), 25 (CEF)
```

And also from the global perspective:

```
CEL6#sh ipv6 traffic | i RPF
               5 RPF drops, 25 RPF suppressed drops
```

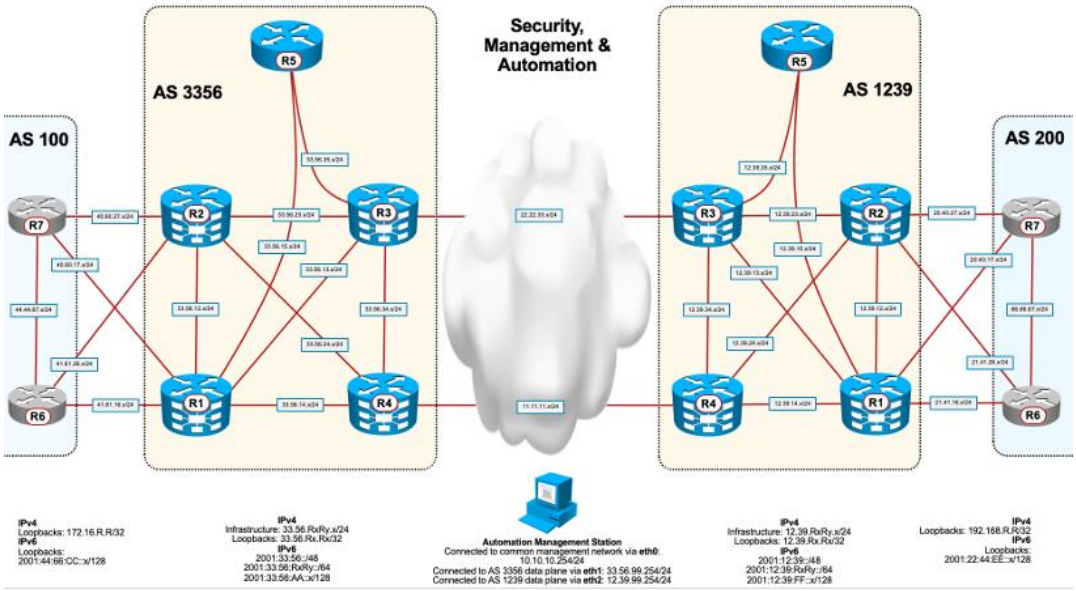
Before continuing, remove the static route from CEL6:

```
CEL6(config)#no ipv6 route 2001:44:66:CC::7/128 Null0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2: BGP RPKI with RPKI server

In our labs, we're running [NLNetLabs Routinator] validating cache server, connected to SP1R5, SP1R4, SP2R5 and SP2R4 routers. We'll use that opportunity to configure and experiment with some of the prefixes.





Note that because of a bug in IOS and IOS-XE RPKI implementation (tracked by CSCvc84848), versions that you may encounter in the exam environment (basically anything before 16.9.6, 16.12.4 and 17.3.2) will fail to properly validate prefixes and at the same time will mark prefixes as Validated ("V" in routing table information) even when there's missing RPKI server connection.

While without using RPKI cache server all configuration is therefore possible (by "faking" some IP that's non-existent RPKI server) we highly recommend getting one of freely available validators in your pod, or use our rack rental services that have option to use RPKI server.

In this lab, we're using 16.9.6 on SP1R5 and SP2R5 to be able to properly work with the RPKI cache server.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2.1: Configure RPKI with IOS-XE connection

First, we'll configure SP2R5 as our RR for AS3356. There's new subnet 33.56.99.0/24 that connects R5, R4 and our RPKI cache server, and cache server is at 33.56.99.1, while R5 is using newly configured 33.56.99.10 address, and R4 is using 33.56.99.11.

```
SP1R5-RR(config)#router bgp 3356
SP1R5-RR(config-router)#bgp rpki server tcp 33.56.99.1 port 3323 refresh 3600
SP1R5-RR(config-router)#bgp bestpath prefix-validate disable
                       ! this is needed to avoid global BGP validation bug
                       ! mentioned above
```

Once connection is established, you can check details of connection itself, as well as number of prefixes received from server:

```
SP1R5-RR#sh bgp ipv4 unicast rpki servers
BGP SOVC neighbor is 33.56.99.1/3323 connected to port 3323
Flags 64, Refresh time is 3600, Serial number is 1, Session ID is 33113
InQ has 0 messages, OutQ has 0 messages, formatted msg 1
Session IO flags 3, Session flags 4008
 Neighbor Statistics:
  Prefixes 249290
  Connection attempts: 1
  Connection failures: 0
  Errors sent: 0
  Errors received: 0

Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: 33.56.99.10, Local port: 16864
Foreign host: 33.56.99.1, Foreign port: 3323
Connection tableid (VRF): 0
Maximum output segment queue size: 50

[…]
```

You'll notice, this sessione exchanges both IPv4 and IPv6 address families. You can check contents of each received RPKI table by executing following requests. For IPv4:

```
SP1R5-RR#sh bgp ipv4 unicast rpki table
190785 BGP sovc network entries using 30525600 bytes of memory
210102 BGP sovc record entries using 6723264 bytes of memory

Network              Maxlen  Origin-AS  Source  Neighbor
1.0.0.0/24           24      13335      0       33.56.99.1/3323
1.0.4.0/24           24      38803      0       33.56.99.1/3323
1.0.4.0/22           22      38803      0       33.56.99.1/3323
1.0.5.0/24           24      38803      0       33.56.99.1/3323
1.0.6.0/24           24      38803      0       33.56.99.1/3323
1.0.7.0/24           24      38803      0       33.56.99.1/3323
[…]
```

For IPv6:

```
SP1R5-RR#sh bgp ipv6 unicast rpki table
36569 BGP sovc network entries using 6728696 bytes of memory
39188 BGP sovc record entries using 1254016 bytes of memory

Network              Maxlen  Origin-AS  Source  Neighbor
2001:200::/32        32      2500       0       33.56.99.1/3323
2001:200:136::/48    48      9367       0       33.56.99.1/3323
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
2001:200:1BA::/48    48      24047      0       33.56.99.1/3323
2001:200:900::/40    40      7660       0       33.56.99.1/3323
2001:200:8000::/35   35      4690       0       33.56.99.1/3323
2001:200:C000::/35   35      23634      0       33.56.99.1/3323
2001:200:E000::/35   35      7660       0       33.56.99.1/3323
2001:218:3002::/48   48      1613       0       33.56.99.1/3323
2001:240::/32        32      2497       0       33.56.99.1/3323
2001:260::/32        48      2518       0       33.56.99.1/3323
2001:288::/32        32      1659       0       33.56.99.1/3323
2001:2F0::/32        128     7514       0       33.56.99.1/3323
2001:300::/32        32      2497       0       33.56.99.1/3323
2001:360::/32        32      135887     0       33.56.99.1/3323
```

As you can see, memory used by overall RPKI subsystem is just subset of full BGP table – overall it's about 45MB for both address families. Memory used is counted as "BGP Router":

```
SP1R5-RR#sh proc memory sorted
Processor Pool Total: 2242554808 Used:  421822456 Free: 1820732352
 lsmpi_io Pool Total:    3149400 Used:    3148568 Free:        832

 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
   0   0 2933349000 2662593784  247839872          0          0 *Init*
 622   0   76074736   76270184   75785184          0          0 BGP Router
 596   0   34427384      51720   34751664          0          0 SBC main process
   0   0   52237144   38633088   13358456    8616303          0 *Dead*
[…]
```

As our RR is not receiving any eBGP paths directly for validation, we have to move to IOS XR based routers to continue with RPKI testing. Before we do that, let's inject couple of prefixes from CEL7 router towards AS 3356.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2.2: Prefix injection towards AS 3356

We'll reconfigure CE-L7 to act as BGP in AS 13335 and advertise three prefixes – two which should validate, and one which will have wrong origin ASN (and should not).

After changes, CE-L7 configuration should look like that:

CE-L7:

```
router bgp 13335
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 40.60.27.2 remote-as 3356
 !
 address-family ipv4
  redistribute static metric 10 route-map STATIC-BGP
  neighbor 40.60.27.2 activate
 exit-address-family
!
ip route 1.0.0.0 255.255.255.0 Null0 tag 13335
ip route 1.1.1.0 255.255.255.0 Null0 tag 13335
ip route 1.6.0.0 255.255.252.0 Null0 tag 9583
!
route-map STATIC-BGP permit 10
 match tag 13335
 set origin igp
 set community 13335:13335
route-map STATIC-BGP permit 20
 match tag 9583
 set origin igp
 set community 9583:9583
!
```

Then, we'll need to adjust SP1R2 configuration to match new BGP session information:

SP1R2:

```
router bgp 3356
 bgp router-id 33.56.2.2
[…]
 !
 neighbor 40.60.27.7
  remote-as 13335
  address-family ipv4 unicast
   route-policy PASS in
   route-policy PASS out
  !
 !
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2.3: Running RPKI with IOS XR connection

We'll configure direct IOS XR session as well. SP1R4 is using 33.56.99.11 to connect to 33.56.99.1:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#router bgp 3356
RP/0/0/CPU0:SP1R4(config-bgp)#rpki server 33.56.99.1
RP/0/0/CPU0:SP1R4(config-bgp-rpki-server)#username server          ! use your
RP/0/0/CPU0:SP1R4(config-bgp-rpki-server)#password password        ! lab values
RP/0/0/CPU0:SP1R4(config-bgp-rpki-server)#transport tcp port 3323  ! for connection
RP/0/0/CPU0:SP1R4(config-bgp-rpki-server)#refresh-time 3600
RP/0/0/CPU0:SP1R4(config-bgp-rpki-server)#response-time 600
RP/0/0/CPU0:SP1R4(config-bgp-rpki-server)#commit
```

Now, we can validate server connectivity summary:

```
RP/0/0/CPU0:SP1R4#sh bgp rpki server summary
Hostname/Address         Transport       State           Time            ROAs (IPv4/IPv6)
33.56.99.1               TCP:3323        ESTAB           00:43:51        210103/39190
```

…current global state of RPKI subsystem:

```
RP/0/0/CPU0:SP1R4#sh bgp rpki summary

RPKI cache-servers configured: 1
RPKI global knobs
  Origin-AS validation is ENABLED globally
  Origin-AS validity WILL NOT affect bestpath selection globally
  Origin-AS validity signaling towards iBGP is DISABLED globally
RPKI database
  Total IPv4 net/path: 190786/210103
  Total IPv6 net/path: 36571/39190
```

…and details of our server connectivity status:

```
RP/0/0/CPU0:SP1R4#sh bgp rpki server 33.56.99.1

RPKI Cache-Server 33.56.99.1
  Transport: TCP port 3323
  Connect state: ESTAB
  Conn attempts: 1
  Total byte RX: 5456624
  Total byte TX: 56
  Last reset
    Timest: Jan 19 21:20:47 (00:47:19 ago)
    Reason: protocol error
RPKI-RTR protocol information
  Serial number: 4
  Cache nonce: 0x8159
  Protocol state: DATA_END
  Refresh  time: 3600 seconds
  Response time: 600 seconds
  Purge time: 60 seconds
  Protocol exchange
    ROAs announced: 210105 IPv4    39194 IPv6
    ROAs withdrawn:      2 IPv4        4 IPv6
    Error Reports :      0 sent        0 rcvd
  Last protocol error
    Reason: received error-report
    Detail: received error-report: no data available
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

We can also check table for IPv4 and IPv6 for prefixes received from the server:

```
RP/0/0/CPU0:SP1R4#show bgp rpki table ipv4

  Network               Maxlen          Origin-AS         Server
  1.0.0.0/24            24              13335             33.56.99.1
  1.0.4.0/22            22              38803             33.56.99.1
  1.0.4.0/24            24              38803             33.56.99.1
  1.0.5.0/24            24              38803             33.56.99.1
  1.0.6.0/24            24              38803             33.56.99.1
  1.0.7.0/24            24              38803             33.56.99.1
  1.1.1.0/24            24              13335             33.56.99.1
  1.1.4.0/22            22              4134              33.56.99.1
  1.1.16.0/20           20              4134              33.56.99.1
  1.2.9.0/24            24              4134              33.56.99.1
[…]

RP/0/0/CPU0:SP1R4#show bgp rpki table ipv6

  Network               Maxlen          Origin-AS         Server
  2001:200::/32         32              2500              33.56.99.1
  2001:200:136::/48     48              9367              33.56.99.1
  2001:200:1ba::/48     48              24047             33.56.99.1
  2001:200:900::/40     40              7660              33.56.99.1
  2001:200:e00::/40     40              4690              33.56.99.1
  2001:200:8000::/35    35              4690              33.56.99.1
  2001:200:c000::/35    35              23634             33.56.99.1
  2001:200:e000::/35    35              7660              33.56.99.1
  2001:218:3002::/48    48              1613              33.56.99.1
  2001:240::/32         32              2497              33.56.99.1
  2001:260::/32         48              2518              33.56.99.1
  2001:288::/32         32              1659              33.56.99.1
  2001:2f0::/32         128             7514              33.56.99.1
  2001:300::/32         32              2497              33.56.99.1
[…]
```

As you may note, BGP prefixes in the table are not verified yet, despite BGP origin-AS validation being enabled:

```
RP/0/0/CPU0:SP1R4#show bgp origin-as validity
[…]

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
Origin-AS validation codes: V valid, I invalid, N not-found, D disabled
    Network            Next Hop            Metric LocPrf Weight Path
 *>i1.0.0.0/24         33.56.2.2               10    100      0 13335 i
 *>i1.1.1.0/24         33.56.2.2               10    100      0 13335 i
 *>i1.6.0.0/22         33.56.2.2               10    100      0 13335 i
 *> 33.56.0.0/16       0.0.0.0                              32768 i
 * i                   33.56.1.1                    100      0 i
 s>i33.56.2.2/32       33.56.3.3                 2  100      0 ?
 s>i33.56.13.0/24      33.56.3.3                 0  100      0 ?
 s> 33.56.14.0/24      0.0.0.0                   0         32768 ?
 s>i33.56.23.0/24      33.56.3.3                 0  100      0 ?
 s> 33.56.24.0/24      0.0.0.0                   0         32768 ?
 s> 33.56.34.0/24      0.0.0.0                   0         32768 ?
 s i                   33.56.3.3                 0  100      0 ?
 s>i33.56.35.0/24      33.56.3.3                 0  100      0 ?
 s> 33.56.99.0/24      0.0.0.0                   0         32768 ?
```

This is because validation only happens on routers that are receiving prefixes over eBGP sessions – which can be further signalled over iBGP.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Let's establish RPKI session directly from R2 to RPKI server. To do that, we have to add 33.56.99.0/24 interface of R4 to OSPF AS3356:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#router ospf AS3356
RP/0/0/CPU0:SP1R4(config-ospf)#area 0
RP/0/0/CPU0:SP1R4(config-ospf-ar)#interface GigabitEthernet0/0/0/3
RP/0/0/CPU0:SP1R4(config-ospf-ar-if)#passive enable
RP/0/0/CPU0:SP1R4(config-ospf-ar-if)#commit
```

And then let's configure RPKI connection from R2:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#rpki server 33.56.99.1
RP/0/0/CPU0:SP1R2(config-bgp-rpki-server)#username nso
RP/0/0/CPU0:SP1R2(config-bgp-rpki-server)#transport tcp port 3323
RP/0/0/CPU0:SP1R2(config-bgp-rpki-server)#refresh-time 3600
RP/0/0/CPU0:SP1R2(config-bgp-rpki-server)#response-time 600
RP/0/0/CPU0:SP1R2(config-bgp-rpki-server)#commit
```

Connection should be up and synchronizing ROAs for couple of seconds:

```
RP/0/0/CPU0:SP1R2#sh bgp rpki server summary

Hostname/Address        Transport       State           Time            ROAs (IPv4/IPv6)
33.56.99.1              TCP:3323        ESTAB           00:01:15        213753/39881

RP/0/0/CPU0:SP1R2#sh bgp rpki server 33.56.99.1
RPKI Cache-Server 33.56.99.1
  Transport: TCP port 3323
  Connect state: ESTAB
  Conn attempts: 1
  Total byte RX: 5551388
  Total byte TX: 20
RPKI-RTR protocol information
  Serial number: 44
  Cache nonce: 0x3717
  Protocol state: DATA_END
  Refresh  time: 3600 seconds
  Response time: 600 seconds
  Purge time: 60 seconds
  Protocol exchange
    ROAs announced: 213754 IPv4    39882 IPv6
    ROAs withdrawn:      0 IPv4        1 IPv6
    Error Reports :      0 sent        0 rcvd
```

Now, the validation should be working correctly:

```
RP/0/0/CPU0:SP1R2#show bgp origin-as validity
[…]
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
Origin-AS validation codes: V valid, I invalid, N not-found, D disabled
    Network            Next Hop            Metric LocPrf Weight Path
V*> 1.0.0.0/24         40.60.27.7              10             0 13335 i
V*> 1.1.1.0/24         40.60.27.7              10             0 13335 i
I*> 1.6.0.0/22         40.60.27.7              10             0 13335 i
[…]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

As R2 is not signalling verification state over iBGP, let's enable that so it can pass the validation info to R4:

```
RP/0/0/CPU0:SP1R2#sh bgp rpki summary
RPKI cache-servers configured: 1
RPKI global knobs
  Origin-AS validation is ENABLED globally
  Origin-AS validity WILL NOT affect bestpath selection globally
  Origin-AS validity signaling towards iBGP is DISABLED globally
RPKI database
  Total IPv4 net/path: 193747/213754
  Total IPv6 net/path: 37191/39881
```

Configure it:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#bgp origin-as validation signal ibgp
RP/0/0/CPU0:SP1R2(config-bgp)#commit
```

Now, also R4 can verify the validity properly:

```
RP/0/0/CPU0:SP1R4#show bgp origin-as validity
[…]
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
Origin-AS validation codes: V valid, I invalid, N not-found, D disabled
    Network            Next Hop            Metric LocPrf Weight Path
V*>i1.0.0.0/24         33.56.2.2                10    100      0 13335 i
 * i                   33.56.2.2                10    100      0 13335 i
V*>i1.1.1.0/24         33.56.2.2                10    100      0 13335 i
 * i                   33.56.2.2                10    100      0 13335 i
I*>i1.6.0.0/22         33.56.2.2                10    100      0 13335 i
 * i                   33.56.2.2                10    100      0 13335 i
[…]
```

Notice one prefix is received via AS3356 Route Reflector (R5) and the other directly via R2. Only path via R2 carries extended community currently, and this is visible both in the above validation report, as well as in details:

```
RP/0/0/CPU0:SP1R4#sh bgp 1.0.0.0/24
[…]
Paths: (2 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  13335
    33.56.2.2 (metric 2) from 33.56.2.2 (33.56.2.2)
      Origin IGP, metric 10, localpref 100, valid, internal, best, group-best
      Received Path ID 0, Local Path ID 0, version 210
      Extended community: VALIDITY:0
      Origin-AS validity: valid (iBGP signalled)
  Path #2: Received by speaker 0
  Not advertised to any peer
  13335
    33.56.2.2 (metric 2) from 33.56.5.5 (33.56.2.2)
      Origin IGP, metric 10, localpref 100, valid, internal
      Received Path ID 0, Local Path ID 0, version 0
      Originator: 33.56.2.2, Cluster list: 33.56.5.5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

And for 1.6.0.0/22 which is obviously in INVALID validation state:

```
RP/0/0/CPU0:SP1R4#sh bgp 1.6.0.0/22
[…]
Paths: (2 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  13335
    33.56.2.2 (metric 2) from 33.56.2.2 (33.56.2.2)
      Origin IGP, metric 10, localpref 100, valid, internal, best, group-best
      Received Path ID 0, Local Path ID 0, version 208
      Extended community: VALIDITY:2
      Origin-AS validity: invalid (iBGP signalled)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2.4: RPKI signalling in IOS-XE

IOS-XE can also signal RPKI validation state over iBGP, and our route-reflector R5, should be already receiving validation state from R2:

```
SP1R5-RR#sh bgp ipv4 unicast
BGP table version is 13, local router ID is 33.56.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
V*>i  1.0.0.0/24       33.56.2.2                10    100      0 13335 i
V*>i  1.1.1.0/24       33.56.2.2                10    100      0 13335 i
I*>i  1.6.0.0/22       33.56.2.2                10    100      0 13335 i
N*>i  33.56.0.0/16     33.56.1.1                      100      0 i
N* i                   33.56.4.4                      100      0 i
N* i                   33.56.2.2                      100      0 i
Nr>i  33.56.2.2/32     33.56.3.3                 2    100      0 ?
Nr>i  33.56.13.0/24    33.56.3.3                 0    100      0 ?
Nr>i  33.56.23.0/24    33.56.3.3                 0    100      0 ?
Nr>i  33.56.34.0/24    33.56.3.3                 0    100      0 ?
Nr>i  33.56.35.0/24    33.56.3.3                 0    100      0 ?
```

To pass the attributes forward (or reflect them), IOS-XE requires additional configuration per peer:

```
SP1R5-RR#conf t
SP1R5-RR(config)#router bgp 3356
SP1R5-RR(config-router)#address-family ipv4 unicast
SP1R5-RR(config-router-af)#neighbor 33.56.1.1 send-community extended
SP1R5-RR(config-router-af)#neighbor 33.56.1.1 announce rpki state
SP1R5-RR(config-router-af)#neighbor 33.56.2.2 send-community extended
SP1R5-RR(config-router-af)#neighbor 33.56.2.2 announce rpki state
SP1R5-RR(config-router-af)#neighbor 33.56.3.3 send-community extended
SP1R5-RR(config-router-af)#neighbor 33.56.3.3 announce rpki state
SP1R5-RR(config-router-af)#neighbor 33.56.4.4 send-community extended
SP1R5-RR(config-router-af)#neighbor 33.56.4.4 announce rpki state
```

Once this configuration is complete, R4 should also receive extended community with validation status passed over from R2 via R5:

```
RP/0/0/CPU0:SP1R4#sh bgp 1.6.0.0/22
[…]
Paths: (2 available, best #1)
[…]
  13335
    33.56.2.2 (metric 2) from 33.56.2.2 (33.56.2.2)
      Origin IGP, metric 10, localpref 100, valid, internal, best, group-best
      Received Path ID 0, Local Path ID 0, version 208
      Extended community: VALIDITY:2
      Origin-AS validity: invalid (iBGP signalled)
  Path #2: Received by speaker 0
  Not advertised to any peer
  13335
    33.56.2.2 (metric 2) from 33.56.5.5 (33.56.2.2)
      Origin IGP, metric 10, localpref 100, valid, internal
      Received Path ID 0, Local Path ID 0, version 0
      Extended community: VALIDITY:2
      Originator: 33.56.2.2, Cluster list: 33.56.5.5
      Origin-AS validity: invalid (iBGP signalled)
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2.5: RPKI policies in IOS-XE

By default, IOS-XE does BGP origin-validation, so the invalid prefix of 1.6/22 won't make it into R2 routing table:

```
SP1R5-RR#sh ip route 1.6.0.0
% Subnet not in table
```

We can adjust this on AF level:

```
SP1R5-RR#conf t
SP1R5-RR(config)#router bgp 3356
SP1R5-RR(config-router)#address-family ipv4 unicast
SP1R5-RR(config-router-af)#bgp bestpath prefix-validate allow-invalid
SP1R5-RR(config-router)#address-family ipv6 unicast
SP1R5-RR(config-router-af)#bgp bestpath prefix-validate allow-invalid
```

And the route will then be passed over from BGP to RIB as local best path:

```
SP1R5-RR#sh ip route 1.6.0.0
Routing entry for 1.6.0.0/22
  Known via "bgp 3356", distance 200, metric 10
  Tag 13335, type internal
  Last update from 33.56.2.2 00:01:15 ago
  Routing Descriptor Blocks:
  * 33.56.2.2, from 33.56.2.2, 00:01:15 ago
      Route metric is 10, traffic share count is 1
      AS Hops 1
      Route tag 13335
      MPLS label: none
```

Other solution is to build a policy using route-maps, that checks validity status and adjusts prefix attributes, or discards prefix altogether:

```
SP1R5-RR(config)#route-map BGP-RPKI permit 10
SP1R5-RR(config-route-map)#match rpki ?
  invalid     RPKI Invalid State
  not-found   RPKI Not Found State
  valid       RPKI Valid State
SP1R5-RR(config-route-map)#match rpki invalid
SP1R5-RR(config-route-map)#set local-preference 10
SP1R5-RR(config-route-map)#route-map BGP-RPKI permit 20
SP1R5-RR(config-route-map)#match rpki not-found
SP1R5-RR(config-route-map)#set local-preference 50
SP1R5-RR(config-route-map)#route-map BGP-RPKI permit 30
SP1R5-RR(config-route-map)#match rpki valid
SP1R5-RR(config-route-map)#set local-preference 150
```

Then, apply policy for ingress prefixes from R2:

```
SP1R5-RR(config-router)#address-family ipv4 unicast
SP1R5-RR(config-router-af)#neighbor 33.56.2.2 route-map BGP-RPKI in
SP1R5-RR#clear ip bgp 33.56.2.2
```

And observe the changed attributes once policy goes into effect:

```
SP1R5-RR#show bgp ipv4 unicast
BGP table version is 21, local router ID is 33.56.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
V*>i  1.0.0.0/24       33.56.2.2                10     150      0 13335 i
V*>i  1.1.1.0/24       33.56.2.2                10     150      0 13335 i
I*>i  1.6.0.0/22       33.56.2.2                10      10      0 13335 i
N* i  33.56.0.0/16     33.56.2.2                               50      0 i
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2.6: RPKI policies with IOS XR

The obvious problem with using that information, is that router by default will continue to forward traffic based on invalid routing information. To change that, we have to change default under address family.

Right now, prefix 1.6.0.0/22 is installed in routing table, as BGP used it to select best path:

```
RP/0/0/CPU0:SP1R4#sh route 1.6.0.0/22
Routing entry for 1.6.0.0/22
  Known via "bgp 3356", distance 200, metric 10
  Tag 13335, type internal
  Installed Jun  4 22:51:19.532 for 00:04:58
  Routing Descriptor Blocks
    33.56.2.2, from 33.56.2.2
      Route metric is 10
  No advertising protos.
```

As in the IOS-XE case, we can make sure INVALID state on prefix immediately discards the prefix from BGP bestpath processing:

```
RP/0/0/CPU0:SP1R4#conf t
RP/0/0/CPU0:SP1R4(config)#router bgp 3356
RP/0/0/CPU0:SP1R4(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R4(config-bgp-af)#bgp bestpath origin-as use validity
RP/0/0/CPU0:SP1R4(config-bgp-af)#commit
```

The effect should be immediate – while the prefix is still visible in BGP, it is no longer candidate for bestpath and therefore, not present in RIB or routing table:

```
RP/0/0/CPU0:SP1R4#sh bgp 1.6.0.0/22 brief
[…]
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
* i1.6.0.0/22         33.56.2.2                10    100      0 13335 i
* i                   33.56.2.2                10     10      0 13335 i
RP/0/0/CPU0:SP1R4#sh route 1.6.0.0/22
% Network not in table
```

Let's reverse the change, and play with IOS XR RPL for RPKI. We will build policy on R2 that checks validation state and acts accordingly:

```
RP/0/0/CPU0:SP1R4(config)#route-policy BGP-RPKI
RP/0/0/CPU0:SP1R4(config-rpl)#if validation-state is ?
  invalid     Invalid State
  not-found   State not found
  valid       Valid State
RP/0/0/CPU0:SP1R4(config-rpl)#if validation-state is invalid then
RP/0/0/CPU0:SP1R4(config-rpl-if)#drop
RP/0/0/CPU0:SP1R4(config-rpl-if)#elseif validation-state is not-found then
RP/0/0/CPU0:SP1R4(config-rpl-elseif)#set local-preference 50
RP/0/0/CPU0:SP1R4(config-rpl-elseif)#elseif validation-state is valid then
RP/0/0/CPU0:SP1R4(config-rpl-elseif)#set local-preference 150
RP/0/0/CPU0:SP1R4(config-rpl-elseif)#endif
RP/0/0/CPU0:SP1R4(config-rpl)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Now, let's apply the policy to neighbor:

```
RP/0/0/CPU0:SP1R2#conf t
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy BGP-RPKI in
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#commit
```

Once policy is applied, INVALID prefix 1.6/22 is no longer in the BGP table on R2 and therefore won't be propagated further into AS 3356:

```
RP/0/0/CPU0:SP1R2#sh bgp ipv4 unicast
[…]
   Network            Next Hop            Metric LocPrf Weight Path
*> 1.0.0.0/24         40.60.27.7               10     150      0 13335 i
*> 1.1.1.0/24         40.60.27.7               10     150      0 13335 i
```

We also don't see it in BGP origin-as validation state check:

```
RP/0/0/CPU0:SP1R2#sh bgp origin-as validity
[…]
Origin-AS validation codes: V valid, I invalid, N not-found, D disabled
    Network            Next Hop            Metric LocPrf Weight Path
V*> 1.0.0.0/24         40.60.27.7               10     150      0 13335 i
V*> 1.1.1.0/24         40.60.27.7               10     150      0 13335 i
 *> 33.56.0.0/16       0.0.0.0                                32768 i
```

Note, neither R5 nor R4 receive 1.6/22 from R2 after applying the policy:

```
SP1R5-RR#sh ip bgp neighbors 33.56.2.2 routes
[…]
     Network          Next Hop            Metric LocPrf Weight Path
V*>i  1.0.0.0/24       33.56.2.2                10     150      0 13335 i
V*>i  1.1.1.0/24       33.56.2.2                10     150      0 13335 i
N* i  33.56.0.0/16     33.56.2.2                       50      0 i
Total number of prefixes 3
```

…and:

```
RP/0/0/CPU0:SP1R4#sh bgp ipv4 unicast neighbors 33.56.2.2 routes
[…]
   Network            Next Hop            Metric LocPrf Weight Path
*>i1.0.0.0/24         33.56.2.2                10     150      0 13335 i
*>i1.1.1.0/24         33.56.2.2                10     150      0 13335 i
* i33.56.0.0/16       33.56.2.2                       100      0 i
Processed 3 prefixes, 3 paths
```

That concludes RPKI tasks.

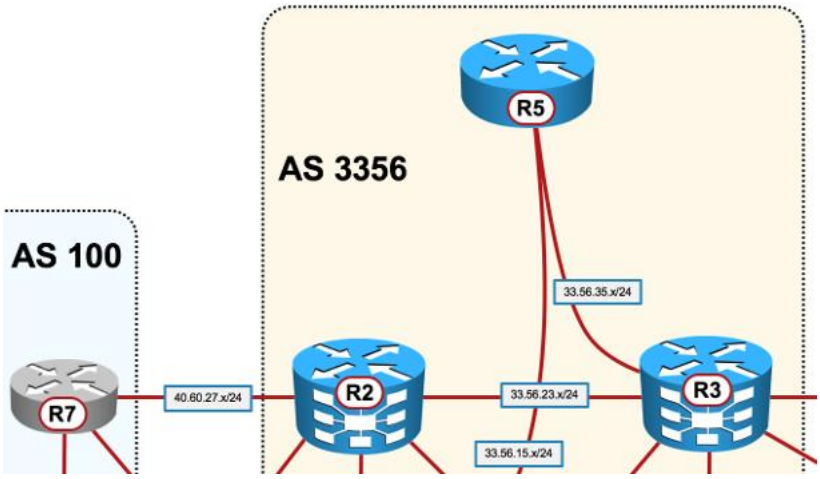
© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4: Deploying SP-wide protection mechanisms

Task 4.1: BGP blackholing

BGP is very efficient tool to distribute NLRIs between neighbors. This allows us to build distributed filter across our BGP domain and protect the network from overflowing data in case of volumetric DDoS.

BGP blackholing is configured on edge routers and requires trigger router – which may be one of the edge routers. We'll use three routers in AS100 and AS3356 to configure example network:



R5 in AS3356 will be our trigger router. Here we'll initiate advertisement to block specific traffic.

R7 in AS100 will be our attacker.

R2 in AS356 will be our edge node that will get advertisements from R5 and block the traffic using blackholing – source and destination.

Task 4.1.1: Configuring trigger router

For R5, we'll define community and trigger route:

```
SP1-R5-RR(config)#ip community-list standard bgp-ddos permit 3356:666
SP1-R5-RR(config)#ip route 192.0.2.1 255.255.255.255 null 0
SP1-R5-RR(config)#ip bgp new-format ! this changes the community view in show commands
to
                                    ! use XXXX:YYYY format rather than single integer
```



There are some parts of the configuration that are not strictly needed on all nodes within network. For example, above 192.0.2.1/32 route is not needed in this specific configuration, however if the below route-map would use 'set ip next-hop 192.0.2.1' (as iBGP sessions propagate next-hops and are not overwritten), that is needed, as BGP will not accept a route with unreachable nexthop for further advertisement.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

That's why it's safer to standardize on the configuration just to make sure everything works as designed.

We'll also define route-map that will be used to redistribute our injected statically routes to BGP for advertisement:

```
SP1-R5-RR(config)#route-map RED-BH permit 10
SP1-R5-RR(config-route-map)#match tag 666
SP1-R5-RR(config-route-map)#set origin igp
SP1-R5-RR(config-route-map)#set local-preference 6666
SP1-R5-RR(config-route-map)#set community 3356:666
SP1-R5-RR(config-route-map)#^Z
```



You may find signalling for BGP blackholing to use not communities but simply next-hop within scope of single AS. This is perfectly fine for single AS, once however organization is willing to share the blackholing information externally, next-hop modification won't work correctly and community is the only way forward.

You can use both or either in singe AS scenario. You'll need to use communities-based signalling in inter-AS scenarios.

We also need to check if router is configured to send communities – this is not default in IOS/IOS-XE:

```
SP1-R5-RR(config)#router bgp 3356
SP1-R5-RR(config-router-af)#neighbor 33.56.1.1 send-community
SP1-R5-RR(config-router-af)#neighbor 33.56.2.2 send-community
SP1-R5-RR(config-router-af)#neighbor 33.56.3.3 send-community
SP1-R5-RR(config-router-af)#neighbor 33.56.4.4 send-community
SP1-R5-RR(config-router-af)#^Z
```

And now, let's configure BGP on R5 to redistribute all routes matching tag we defined – 666:

```
SP1-R5-RR(config)#router bgp 3356
SP1-R5-RR(config-router)#address-family ipv4 unicast
SP1-R5-RR(config-router-af)#redistribute static route-map RED-BH
SP1-R5-RR(config-router-af)#^Z
```

Let's check if the redistribution works by adding one additional test route:

```
SP1-R5-RR(config)#ip route 192.0.2.254 255.255.255.255 null 0 tag 666
```

And now, let's verify BGP:

```
SP1-R5-RR#sh ip bgp 192.0.2.254
BGP routing table entry for 192.0.2.254/32, version 28
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (33.56.5.5)
      Origin IGP, metric 0, localpref 6666, weight 32768, valid, sourced, best
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
      Community: 3356:666
      rx pathid: 0, tx pathid: 0x0
```

Let's verify if we're receiving the route on SP1R2:

```
RP/0/0/CPU0:SP1R2#sh ip bgp 192.0.2.254
BGP routing table entry for 192.0.2.254/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 35          35
Last Modified: Jan 01 16:12:28.762 for 00:00:21
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    40.60.27.7
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    40.60.27.7
  Local
    33.56.5.5 (metric 3) from 33.56.5.5 (33.56.5.5)
      Origin IGP, metric 0, localpref 6666, valid, internal, best, group-best
      Received Path ID 0, Local Path ID 0, version 35
      Community: 3356:666
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.1.2: Configuring edge router

On all edge routers we need to make sure there's a Null0 route and router based on community value properly modifies next-hop in incoming NLRIs to this Null0 route.

First, we'll define Null0 route:

```
RP/0/0/CPU0:SP1R2(config)#router static
RP/0/0/CPU0:SP1R2(config-static)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-static-afi)#192.0.2.1/32 null 0
RP/0/0/CPU0:SP1R2(config-static-afi)#commit
```

Now, we'll define a policy for BGP to match the community and properly set next-hop – we can either use direct 'discard', or point to 192.0.2.1 which recursively also points to Null0.

```
RP/0/0/CPU0:SP1R2(config)#route-policy RTBH
RP/0/0/CPU0:SP1R2(config-rpl)#if community matches-any (3356:666) then
RP/0/0/CPU0:SP1R2(config-rpl-if)#set next-hop discard
RP/0/0/CPU0:SP1R2(config-rpl-if)#else
RP/0/0/CPU0:SP1R2(config-rpl-else)#pass
RP/0/0/CPU0:SP1R2(config-rpl-else)#endif
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
```

Now, let's modify the session to route reflector to properly adjust the policy:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 33.56.5.5
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy RTBH in
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

The route we received now correctly points directly to Null0 adjacency:

```
RP/0/0/CPU0:SP1R2#sh cef 192.0.2.254
192.0.2.254/32, version 99, null0 adjacency, internal 0x5000001 0x0 (ptr 0xa14091f4)
[1], 0x0 (0xa13d3710), 0x0 (0x0)
 Updated Jan 01 16:18:16.276
 Prefix Len 32, traffic index 0, precedence n/a, priority 4
   via 0.0.0.0/32, Null0, 5 dependencies, weight 0, class 0 [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa0d40050 0x0]
    next hop 0.0.0.0/32
     null0 adjacency
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.1.3: “Starting the DDoS”

As the route is propagated to AS100 by BGP, we don’t need to explicitly point CEL7 to SP1R2 to start generating the traffic:

```
CEL7#sh ip route 192.0.2.254
Routing entry for 192.0.2.254/32
  Known via "bgp 100", distance 20, metric 0
  Tag 3356, type external
  Last update from 40.60.27.2 00:11:32 ago
  Routing Descriptor Blocks:
  * 40.60.27.2, from 40.60.27.2, 00:11:32 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 3356
      MPLS label: none
```

We’ll ping the address to test the protection:

```
CEL7#ping 192.0.2.254 size 1500 repeat 5000
Type escape sequence to abort.
Sending 5000, 1500-byte ICMP Echos to 192.0.2.254, timeout is 2 seconds:
..
```

Now, let’s verify on SP1R2:

```
RP/0/0/CPU0:SP1R2#sh cef drops | i Null0
Null0 drops          packets :              47
```



This configuration reflects what’s called RTBH – for Remote-Triggered Blackholing. When You add uRPF to the edge interfaces, You’ll also drop traffic coming FROM addresses distributed in BGP which is called S/RTBH (Source Remote-Triggered Blackholing).

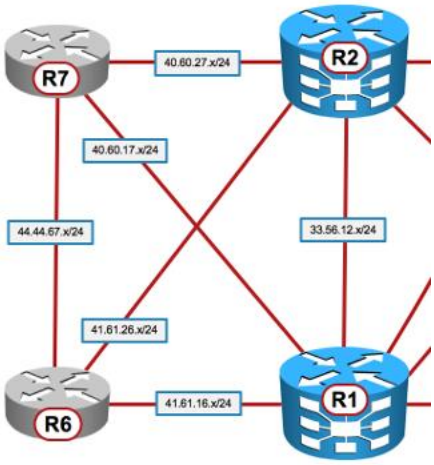
uRPF configuration was covered before in the lab.

Remove all the configuration before moving to the next task.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.2: BGP FlowSpec

To experiment with BGP FlowSpec AF, we'll use the same topology but in a different setup. R2 will be our trigger router (running IOS XR) and R7 (IOS-XE) will be our edge router protecting the network from attack coming in from R6.





XRv only supports server mode (no option to invoke FlowSpec interface attachment) so we'll use IOS XR to inject flow specification and IOS-XE to block traffic. You can use also other tools in real life to inject FlowSpec – like for example ExaBGP.

Task 4.2.1: Configuring trigger router (FlowSpec server)

FlowSpec in BGP requires its own address-family:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 flowspec
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy PASS in
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#route-policy PASS out
```

We'll define traffic to announce via FlowSpec and then announce it in FlowSpec:

```
RP/0/0/CPU0:SP1R2(config)#class-map type traffic match-all FLOW1
RP/0/0/CPU0:SP1R2(config-cmap)#match source-address ipv4 172.16.6.6/32
RP/0/0/CPU0:SP1R2(config-cmap)#match destination-port 23
RP/0/0/CPU0:SP1R2(config-cmap)#exit
RP/0/0/CPU0:SP1R2(config)#policy-map type pbr PB-FS
RP/0/0/CPU0:SP1R2(config-pmap)#class type traffic FLOW1
RP/0/0/CPU0:SP1R2(config-pmap-c)#drop
RP/0/0/CPU0:SP1R2(config-pmap-c)#exit
RP/0/0/CPU0:SP1R2(config-pmap)#class-map type traffic match-all FLOW1
RP/0/0/CPU0:SP1R2(config-cmap)#exit
RP/0/0/CPU0:SP1R2(config)#flowspec
RP/0/0/CPU0:SP1R2(config-flowspec)#address-family ipv4
RP/0/0/CPU0:SP1R2(config-flowspec-af)#service-policy type pbr PB-FS
RP/0/0/CPU0:SP1R2(config-flowspec-af)#commit
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.2.1: Configuring edge router (FlowSpec client)

On our edge devices we'll also activate FlowSpec AF on session to route reflector:

```
CEL7(config)#router bgp 100
CEL7(config-router)#address-family ipv4 flowspec
CEL7(config-router-af)#neighbor 40.60.27.2 activate
```

Session should be properly established before we go further:

```
CEL7(config)#flowspec
CEL7(config-flowspec)#address-family ipv4
CEL7(config-flowspec-afi)#local-install interface-all
CEL7(config-flowspec-afi)#address-family ipv6
CEL7(config-flowspec-afi)#local-install interface-all
```

On CEL7 we should get advertisement from FS:

```
CEL7#show bgp ipv4 flowspec summary
[…]
BGP activity 26/8 prefixes, 94/63 paths, scan interval 60 secs
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
40.60.27.2      4         3356      15      21        2    0    0 00:06:14        1

CEL7#show bgp ipv4 flowspec
BGP table version is 2, local router ID is 172.16.7.7
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>   Source:172.16.6.6/32,DPort:=23
                      0.0.0.0                                0 3356 i
```

We can also check detailed route information:

```
CEL7#show bgp ipv4 flowspec detail
BGP routing table entry for Source:172.16.6.6/32,DPort:=23, version 2
  Paths: (1 available, best #1, table IPv4-Flowspec-BGP-Table)
  […]
  3356
    0.0.0.0 from 40.60.27.2 (33.56.2.2)
      Origin IGP, localpref 100, valid, external, best
      Extended Community: FLOWSPEC Traffic-rate:3356,0
      rx pathid: 0, tx pathid: 0x0
```

Once we'll initiate traffic to port 23/tcp from CEL7 loopback0 interface:

```
CEL6#telnet 172.16.7.7 /source-interface loopback 0
Trying 172.16.7.7 ...
% Connection timed out; remote host not responding
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

You can then check the drops according to FlowSpec policy:

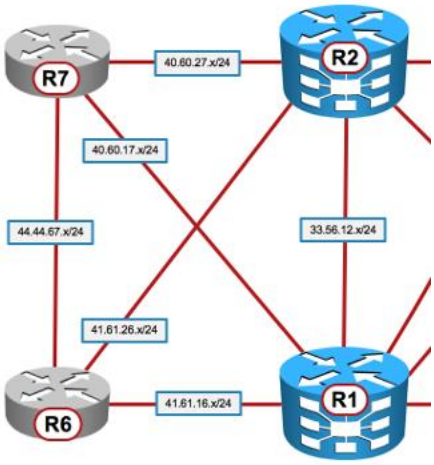
```
CEL7#show flowspec ipv4 detail
AFI: IPv4
  Flow           :Source:172.16.6.6/32,DPort:=23
    Actions      :Traffic-rate: 0 bps  (bgp.1)
    Statistics                          (packets/bytes)
      Matched             :                    4/232
      Dropped             :                    4/232
```

Please remove all FS configuration from both peers before continuing.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4.3: QoS Policy Propagation via BGP (QPPB)

For QPPB configuration we'll use the same basic topology and configuration as in previous task. We'll announce traffic to be policed from R2 to R7, and on R7 we'll apply the QPPB configuration to Gi3 interface connecting R7 and R6:



Task 4.3.1: Configuring trigger router

We'll configure redistribution policy on R2 to match for tag 700 and then apply specific community (3356:700) to routes:

```
RP/0/0/CPU0:SP1R2(config)#route-policy RED-QPPB
RP/0/0/CPU0:SP1R2(config-rpl)#if tag in ( 700 ) then
RP/0/0/CPU0:SP1R2(config-rpl-if)#set community ( 3356:700 )
RP/0/0/CPU0:SP1R2(config-rpl-if)#else
RP/0/0/CPU0:SP1R2(config-rpl-else)#pass
RP/0/0/CPU0:SP1R2(config-rpl-else)#endif
RP/0/0/CPU0:SP1R2(config-rpl)#end-policy
```

Next, we'll define static route to be redistributed:

```
RP/0/0/CPU0:SP1R2(config)#router static
RP/0/0/CPU0:SP1R2(config-static)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-static-afi)#192.0.2.77/32 null 0 tag 700
```

And now we'll configure the redistribution itself:

```
RP/0/0/CPU0:SP1R2(config-bgp)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-af)#redistribute static route-policy RED-QPPB
RP/0/0/CPU0:SP1R2(config-bgp-af)#commit
```

Let's check the redistributed route along with its communities:

```
RP/0/0/CPU0:SP1R2#sh bgp ipv4 unicast 192.0.2.77/32
BGP routing table entry for 192.0.2.77/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker               120         120
Last Modified: Jan 01 18:27:13.762 for 00:00:40
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    33.56.5.5       40.60.27.7
  Path #1: Received by speaker 0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Advertised to peers (in unique update groups):
    33.56.5.5       40.60.27.7
  Local
    0.0.0.0 from 0.0.0.0 (33.56.2.2)
      Origin incomplete, metric 0, localpref 100, weight 32768, valid, redistributed,
best, group-best
      Received Path ID 0, Local Path ID 0, version 120
      Community: 3356:700
```

The only thing left to do on SP1R2 is to enable sending of communities towards CEL7:

```
RP/0/0/CPU0:SP1R2(config)#router bgp 3356
RP/0/0/CPU0:SP1R2(config-bgp)#neighbor 40.60.27.7
RP/0/0/CPU0:SP1R2(config-bgp-nbr)#address-family ipv4 unicast
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#send-community-ebgp
RP/0/0/CPU0:SP1R2(config-bgp-nbr-af)#commit
```

We're now moving to CEL7 and checking if the route was received:

```
CEL7#sh bgp ipv4 unicast 192.0.2.77
BGP routing table entry for 192.0.2.77/32, version 16
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     11
  Refresh Epoch 1
  3356
    172.16.6.6 (metric 301) from 172.16.6.6 (172.16.6.6)
      Origin incomplete, metric 0, localpref 100, valid, internal
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  3356
    40.60.27.2 from 40.60.27.2 (33.56.2.2)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      Community: 3356:700
      rx pathid: 0, tx pathid: 0x0
```

Now, let's build policy that will match prefix with QoS group and then enforce limits on interface Gi3:

```
CEL7(config)#ip community-list 100 permit 3356:700
CEL7(config)#route-map RM-QPPB permit 10
CEL7(config-route-map)#match community 100
CEL7(config-route-map)#set ip qos-group 66
```

This policy needs to be applied under IPv6 AF definition of BGP process:

```
CEL7(config)#router bgp 100
CEL7(config-router)#address-family ipv4
CEL7(config-router-af)#table-map RM-QPPB
```

We have to define QoS policy that will match QoS-group 66 and then apply some policies to traffic matching it. Our simplified config will drop both conforming and exceeding traffic as illustration – in real world QPPB is generally used to police traffic down without dropping all of it – for that, blackholing would be much more efficient:

```
CEL7(config-if)#class-map CM-QoS-QPPB
CEL7(config-cmap)#match qos-group 66
CEL7(config)#policy-map PM-QoS-QPPB
CEL7(config-pmap)#class CM-QoS-QPPB
CEL7(config-pmap-c)#police 8192 conform-action drop exceed-action drop
CEL7(config-pmap-c-police)#^Z
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Now, let's configure Gi3 for QPPB:

```
CEL7(config-route-map)#int gi3
CEL7(config-if)#bgp-policy destination ip-qos-map
CEL7(config-if)#service-policy input PM-QoS-QPPB
```

Before we continue, we have to force traffic to 192.0.2.77/32 from CEL6 to go via CEL7 and not directly to SP1R2 (as the CEL6 is also connected to SP1R2 and receives the same IPv6 prefix, albeit without community).

Let's go nuclear and shut down IPv4 AF on BGP configuration for CEL6:

```
CEL6(config)#router bgp 100
CEL6(config-router)#address-family ipv4
CEL6(config-router-af)# no neighbor 41.61.16.1 activate
CEL6(config-router-af)#^Z
```

Now, CEL6 should route all traffic to 192.0.2.77 via CEL7:

```
CEL6#sh ip route 192.0.2.77
Routing entry for 192.0.2.77/32
  Known via "bgp 100", distance 200, metric 0
  Tag 3356, type internal
  Last update from 172.16.7.7 00:00:26 ago
  Routing Descriptor Blocks:
  * 172.16.7.7, from 172.16.7.7, 00:00:26 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 3356
      MPLS label: none
```

Let's generate some traffic to the prefix:

```
CEL6#ping 192.0.2.77 size 1500 repeat 100 timeout 0
Type escape sequence to abort.
Sending 100, 1500-byte ICMP Echos to 192.0.2.77, timeout is 2 seconds:
...
```

Prefix 192.0.2.77 should belong to QoS group 66 on CEL7:

```
CEL7#sh ip cef 192.0.2.77 detail
192.0.2.77/32, epoch 2, flags [rib only nolabel, rib defined all labels]
  QOS: qos-group 66
  recursive via 40.60.27.2
    attached to GigabitEthernet2
```

…and traffic should be properly classified to our class-map on Gi3 interface:

```
CEL7#sh policy-map interface gigabitEthernet 3
 GigabitEthernet3

  Service-policy input: PM-QoS-QPPB

    Class-map: CM-QoS-QPPB (match-all)
      40 packets, 45016 bytes
      30 second offered rate 6000 bps, drop rate 6000 bps
      Match: qos-group 66
      police:
          cir 8000 bps, bc 1500 bytes
        conformed 11 packets, 1110 bytes; actions:
          drop
        exceeded 29 packets, 43906 bytes; actions:
          drop
        conformed 0000 bps, exceeded 6000 bps
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
  Class-map: class-default (match-any)
    2138 packets, 231366 bytes
    30 second offered rate 0000 bps, drop rate 0000 bps
    Match: any
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 5: Other system-related options

Task 5.1: Benchmarking/testing link between routers

Cisco routers offer built-in TTCP feature. While the process works on router RP and you shouldn't expect too much of performance, it's nice tool to generate some traffic over the link (more than with typical ping), or congestion while testing security and/or QoS features.

On destination IOS XR host run following command:

```
RP/0/0/CPU0:SP1R1#ttcp receive source
ttcp-r: thread = 1, buflen=8192, nbuf=2048, align=16384/0, port=5001  tcp
ttcp-r: socket
```

On source IOS XR host run following command, pointing TTCP to first router (usually, it's loopback interface unless you want to test some specific link – we'll use loopback for this test):

```
RP/0/0/CPU0:SP1R2#ttcp transmit source host 33.56.1.1
ttcp-t: thread = 1, buflen=8192, nbuf=2048, align=16384/0, port=5001  tcp  -> 33.56.1.1
ttcp-t: socket
ttcp-t: connect
```

After a while, source will finish the test with default settings and report the test results:

```
TTCP: +++ all threads terminated +++
ttcp-t: 16777216 bytes in 13.67 real useconds = 1198.62 KB/sec +++
ttcp-t: 2048 I/O calls, msec/call = 6.83, calls/sec = 149.83
ttcp-t:
```

The destination will also break out from console TTCP process and report result:

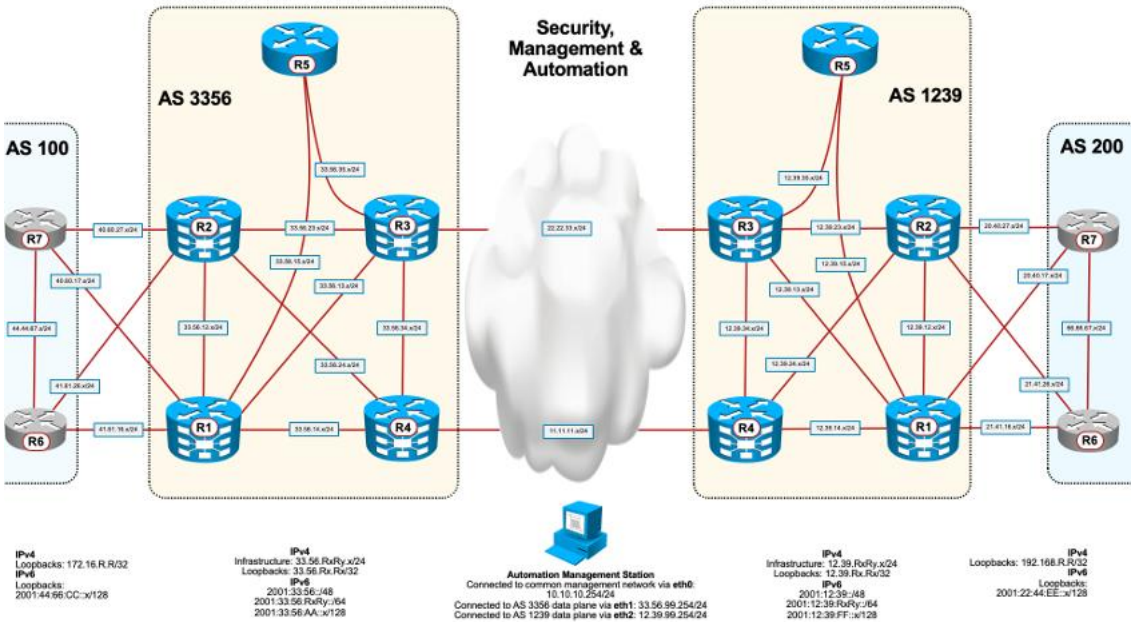
```
TTCP: +++ all threads terminated +++
ttcp-r: 16777216 bytes in 13.68 real useconds = 1197.74 KB/sec +++
ttcp-r: 13475 I/O calls, msec/call = 1.04, calls/sec = 985.08
ttcp-r:
```

So, it seems that our virtual routers, limited to control-plane processing, were able to push 1.1MB/s over the link connecting them. Quite a result ☺

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Chapter 19: Network automation in SP networks

For system and network we'll use following topology:



This chapter uses following init files:

- **For VIRL/CML 1.x:** `sp.bootcamp.ch19.virl1.virl`
- **For VIRL^2/CML^2:** `sp.bootcamp.ch19.virl2.yaml`
- **For EVE-NG:** `CCIE SP bootcamp ch19.unl`



Note, that the configuration file calls for node that you need to build yourself. It's basic Linux Ubuntu 20 server, with following packages installed:

- Ansible
- Routinator
- NSO

Detailed instructions to build such image are outside of scope of this workbook.

If you need to rent ready-made pod aligned with this chapter configuration, please contact us directly.

Make sure your server has three network interfaces, defined as follows:

- **eth0**, connected to bridged segment with router management interfaces (Gi1 for IOS-XE based routers, and MgmtEth0/0/CPU0/0 for IOS XR based routers):

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

    - IPv4: 10.10.10.254, interface is defined with a mask of /24
    - IPv6: 2001:10:10::254/64
- **eth1**, connected to bridged segment in AS 3356 with R4 and R5:
    - IPv4: 33.56.99.254/24
    - IPv6: 2001:33:56::254/64
- **eth2**, connected to bridged segment in AS 1239 with R4 and R5:
    - IPv4: 12.39.99.254/24
    - IPv6: 2001:12:39::254/64

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1: Basic Ansible concepts

Task 1.1: Installing & verifying Ansible

Ansible is great automation tool, usable in many different scenarios.

Your host should have Ansible package already installed. If needed, install it before continuing. While it's outside of the scope of the workbook, simple sequence should work for Linux-based OS:

```
~$ sudo apt install ansible
[…]
Need to get 9643 kB of archives.
After this operation, 90.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
[…]
Setting up python3-winrm (0.3.0-2) ...
Setting up ansible (2.9.6+dfsg-1) ...
Processing triggers for man-db (2.9.1-1) ...
~$ sudo apt install python3-paramiko
```

Task 1.2: Initial Ansible configuration

First, let's create `/etc/ansible/hosts` file, which will hold our inventory. You may use per-project specific inventory, or pass inventory with each Ansible invocation, but that setup works for single-use deployments like in your pod.

Below is sample of how the file should be built. Of course, reference your own routers in your own pod. Bootcamp git repository also contains the full example file.

```
[xe:vars]
ansible_network_os=ios
ansible_user=cisco
ansible_password=cisco
ansible_connection=network_cli

[xr:vars]
ansible_network_os=iosxr
ansible_user=cisco
ansible_password=cisco
ansible_connection=network_cli

[xe]
SP1R5 ansible_host=10.10.10.15

[xr]
SP1R1 ansible_host=10.10.10.11
SP1R2 ansible_host=10.10.10.12
SP1R3 ansible_host=10.10.10.13
SP1R4 ansible_host=10.10.10.14
```

Before we run first playbook, we need to either turn off host key checking (for SSH) or log in from the server you're working on to every switch so we can get their fingerprints. Turning off host key checking is faster and simpler, but that's not solution for production environments 😉

Create `ansible.cfg` in the directory you'll run playbooks from and put there:

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
[defaults]
host_key_checking = False
```

We also have to generate keys for IOS XR virtual routers. Make sure you're using version that supports full k9 capabilities (old demo 6.1.x images don't have this capability). In this lab we're using 6.5.3:

```
RP/0/0/CPU0:SP2R3#crypto key generate rsa
The name for the keys will be: the_default
  Choose the size of the key modulus in the range of 512 to 4096 for your General
Purpose Keypair. Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus [2048]: <ENTER>
Generating RSA keys ...
Done w/ crypto generate keypair
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.3: First Ansible playbook

Now, we can run very simple playbook and check if we have proper communication with the routers (save below in `router-backup.yml`):

```
---

- name: "XE routers connection"
  hosts: xe
  gather_facts: no

  tasks:
    - name: Backup XE routers configuration
      ios_config:
        backup: yes

- name: "XR routers connection"
  hosts: xr
  gather_facts: no

  tasks:
    - name: Backup XR routers configuration
      ios_xr_config:
        backup: yes
```

Once that's complete, you can run your first playbook:

```
~$ ansible-playbook -i /etc/ansible/hosts -v router-backup.yml

PLAY [XE routers connection]
*********************************************************************

TASK [Backup XE routers configuration]
*****************************************************
changed: [SP1R5] => {"ansible_facts": {"discovered_interpreter_python":
"/usr/bin/python"}, "backup_path": "/home/nso/backup/SP1R5_config.2021-01-06@11:01:59",
"changed": true, "date": "2021-01-06", "filename": "SP1R5_config.2021-01-06@11:01:59",
"shortname": "/home/nso/backup/SP1R5_config", "time": "11:01:59"}

PLAY [XR routers connection]
*********************************************************************

TASK [Backup XR routers configuration]
*****************************************************
changed: [SP1R3] => {"ansible_facts": {"discovered_interpreter_python":
"/usr/bin/python"}, "backup_path": "/home/nso/backup/SP1R3_config.2021-01-06@11:02:04",
"changed": true, "date": "2021-01-06", "filename": "SP1R3_config.2021-01-06@11:02:04",
"shortname": "/home/nso/backup/SP1R3_config", "time": "11:02:04"}
changed: [SP1R2] => {"ansible_facts": {"discovered_interpreter_python":
"/usr/bin/python"}, "backup_path": "/home/nso/backup/SP1R2_config.2021-01-06@11:02:04",
"changed": true, "date": "2021-01-06", "filename": "SP1R2_config.2021-01-06@11:02:04",
"shortname": "/home/nso/backup/SP1R2_config", "time": "11:02:04"}
changed: [SP1R4] => {"ansible_facts": {"discovered_interpreter_python":
"/usr/bin/python"}, "backup_path": "/home/nso/backup/SP1R4_config.2021-01-06@11:02:04",
"changed": true, "date": "2021-01-06", "filename": "SP1R4_config.2021-01-06@11:02:04",
"shortname": "/home/nso/backup/SP1R4_config", "time": "11:02:04"}
changed: [SP1R1] => {"ansible_facts": {"discovered_interpreter_python":
"/usr/bin/python"}, "backup_path": "/home/nso/backup/SP1R1_config.2021-01-06@11:02:04",
"changed": true, "date": "2021-01-06", "filename": "SP1R1_config.2021-01-06@11:02:04",
"shortname": "/home/nso/backup/SP1R1_config", "time": "11:02:04"}

PLAY RECAP
*********************************************************************************
SP1R1    : ok=1  changed=1 unreachable=0  failed=0  skipped=0  rescued=0    ignored=0
SP1R2    : ok=1  changed=1 unreachable=0  failed=0  skipped=0  rescued=0    ignored=0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
SP1R3     : ok=1   changed=1 unreachable=0   failed=0   skipped=0   rescued=0     ignored=0
SP1R4     : ok=1   changed=1 unreachable=0   failed=0   skipped=0   rescued=0     ignored=0
SP1R5     : ok=1   changed=1 unreachable=0   failed=0   skipped=0   rescued=0     ignored=0
```

As you can see, backup configuration files were created under /backup/ directory in your current working config. You can list them and view if needed:

```
~$ ls -la backup/*
-rw-rw-r-- 1 nso nso 5488 Jan  6 11:02 backup/SP1R1_config.2021-01-06@11:02:04
-rw-rw-r-- 1 nso nso 6642 Jan  6 11:02 backup/SP1R2_config.2021-01-06@11:02:04
-rw-rw-r-- 1 nso nso 5456 Jan  6 11:02 backup/SP1R3_config.2021-01-06@11:02:04
-rw-rw-r-- 1 nso nso 5567 Jan  6 11:02 backup/SP1R4_config.2021-01-06@11:02:04
-rw-rw-r-- 1 nso nso 9746 Jan  6 11:01 backup/SP1R5_config.2021-01-06@11:01:59
```

Note this playbook will create backup of your routers configuration in the “backup” subdirectory. Each time it’ll be new backup as time goes on, hence “changed” in the column for each switch.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 1.4: Running Ansible playbooks in debug mode

Next, we can ask routers to execute some basic commands and then review the contents or act on specific content returned. Here, we're going to use this simple playbook to gather output of 'show version':

```
nso@nso:~$ more pod-system-version.yml
---

- name: "IOS XR show version execution"
  hosts: xr
  gather_facts: no

  tasks:
   - name: run show version on remote devices
     iosxr_command:
       commands: show version

- name: "IOS XE show version execution"
  hosts: xe
  gather_facts: no

  tasks:
    - name: run show version on remote devices
      ios_command:
        commands: show version
```

With extra debug options passed to Ansible (`-vv`) we'll going to review what's being returned:

```
nso@nso:~$ ansible-playbook -vv pod-system-version.yml
ansible-playbook 2.9.6
  config file = /home/nso/ansible.cfg
  configured module search path = ['/home/nso/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible-playbook
  python version = 3.8.5 (default, Jan 27 2021, 15:41:15) [GCC 9.3.0]
Using /home/nso/ansible.cfg as config file

PLAYBOOK: pod-system-version.yml
****************************************************************************
2 plays in pod-system-version.yml

PLAY [IOS XR show version execution]
****************************************************************************
META: ran handlers

TASK [run show version on remote devices]
****************************************************************
task path: /home/nso/pod-system-version.yml:8
ok: [SP1R1] => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python"},
"changed": false, "stdout": ["Cisco IOS XR Software, Version 6.5.3[Default]\nCopyright
(c) 2019 by Cisco Systems, Inc.\n\nROM: GRUB, Version 1.99(0), DEV RELEASE\n\nSP1R1
uptime is 1 day, 2 hours, 11 minutes\nSystem image file is \"bootflash:disk0/xrvr-os-
mbi-6.5.3/mbixrvr-rp.vm\"\n\ncisco IOS XRv Series (Pentium Celeron Stepping 3) processor
with 3145215K bytes of memory.\nPentium Celeron Stepping 3 processor at 2343MHz,
Revision 2.174\nIOS XRv Chassis\n\n1 Management Ethernet\n6 GigabitEthernet\n97070k
bytes of non-volatile configuration memory.\n866M bytes of hard disk.\n2321392k bytes of
disk0: (Sector size 512 bytes).\n\nConfiguration register on node 0/0/CPU0 is
0x2102\nBoot device on node 0/0/CPU0 is disk0:\nPackage active on node 0/0/CPU0:\niosxr-
security, V 6.5.3[Default], Cisco Systems, at disk0:iosxr-security-6.5.3\n    Built on
Tue Mar 26 13:57:09 UTC 2019\n
[…]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
PLAY [IOS XE show version execution]
*********************************************************************
META: ran handlers

TASK [run show version on remote devices]
*********************************************************************
task path: /home/nso/pod-system-version.yml:18
ok: [SP1R5] => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python"},
"changed": false, "stdout": ["Cisco IOS XE Software, Version 16.09.06\nCisco IOS
Software [Fuji], Virtual XE Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version 16.9.6,
RELEASE SOFTWARE (fc2)\nTechnical Support: http://www.cisco.com/techsupport\nCopyright
[...]

PLAY RECAP
*********************************************************************************************
SP1R1      : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
SP1R2      : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
SP1R3      : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
SP1R4      : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
SP1R5      : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2: Advanced Ansible concepts

Task 2.1: Introducing configuration changes using Ansible

Let's do some basic reconfiguration of our routers.



It's fair on the exam to ask you to do something specific in Ansible on your devices. Likely, some configuration will be already present, so as always, it's good to read exactly the specific requirement and examine your existing environment.

To see all modules for IOS-XE or IOS XR, refer to Ansible documentation, that should also be available during exam:

- https://docs.ansible.com/ansible/2.9/modules/list_of_network_modules.html#iosxr
- https://docs.ansible.com/ansible/2.9/modules/list_of_network_modules.html#ios

First, we'll make sure all of our routers have domain-name set to 'netdesign.zone' and name servers configured to reach to Umbrella DNS servers (208.67.222.222 and 208.67.220.220):

```
nso@nso:~$ more pod-system-conf.yml
---

- name: "IOS XR routers configuration"
  hosts: xr
  gather_facts: no

  tasks:
   - name: configure domain-name
     iosxr_system:
      domain_name: netdesign.zone

   - name: configure name servers
     iosxr_system:
      name_servers:
       - 208.67.222.222
       - 208.67.220.220

- name: "IOS XE routers configuration"
  hosts: xe
  gather_facts: no

  tasks:
    - name: configure domain-name
      ios_system:
       domain_name: netdesign.zone

    - name: configure name servers
      ios_system:
       name_servers:
        - 208.67.222.222
        - 208.67.220.220
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

This once executed should result in following output:

```
nso@nso:~$ ansible-playbook pod-system-conf.yml

PLAY [IOS XR routers configuration]
********************************************************************************

TASK [configure domain-name]
****************************************************
changed: [SP1R2]
changed: [SP1R3]
changed: [SP1R4]
changed: [SP1R1]

TASK [configure name servers]
*************************************************************************************
changed: [SP1R2]
changed: [SP1R1]
changed: [SP1R3]
changed: [SP1R4]

PLAY [IOS XE routers configuration]
********************************************************************************

TASK [configure domain-name]
*************************************************************************************
ok: [SP1R5]

TASK [configure name servers]
*************************************************************************************
changed: [SP1R5]

PLAY RECAP
*************************************************************************************************
************
SP1R1    : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
SP1R2    : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
SP1R3    : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
SP1R4    : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
SP1R5    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

And routers should have configuration adjusted accordingly:

```
RP/0/0/CPU0:SP1R2#sh running-config | i domain
domain name netdesign.zone
domain name-server 208.67.222.222
domain name-server 208.67.220.220
SP1R5-RR#sh running-config | i name
hostname SP1R5-RR
ip name-server 208.67.222.222 208.67.220.220
ip domain name netdesign.zone
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 2.2: Monitoring reachability using Ansible

As you can execute commands on hosts/routers, you can also perform basic connectivity checks and potentially react to lack of connectivity.

Here's simple ping script to check availability of Management Station (10.10.10.254) over at VRF Mgmt-intf from each and every router:

```
nso@nso:~$ more pod-ping-management.yml
---

- name: "IOS XR ping management station"
  hosts: xr
  gather_facts: no

  tasks:
   - name: ping management station in VRF
     iosxr_command:
       commands: ping vrf Mgmt-intf 10.10.10.254

- name: "IOS XE ping management station"
  hosts: xe
  gather_facts: no

  tasks:
    - name: ping management station in VRF
      ios_command:
        commands: ping vrf Mgmt-intf 10.10.10.254
```

And here's execution of the playbook with debugging of responses:

```
nso@nso:~$ ansible-playbook -vv pod-ping-management.yml
ansible-playbook 2.9.6
  config file = /home/nso/ansible.cfg
  configured module search path = ['/home/nso/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible-playbook
  python version = 3.8.5 (default, Jan 27 2021, 15:41:15) [GCC 9.3.0]
Using /home/nso/ansible.cfg as config file

PLAYBOOK: pod-ping-management.yml
****************************************************************************
2 plays in pod-ping-management.yml

PLAY [IOS XR ping management station]
****************************************************************************
META: ran handlers

TASK [ping management station in VRF]
****************************************************************************
task path: /home/nso/pod-ping-management.yml:8
ok: [SP1R4] => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python"},
"changed": false, "stdout": ["Type escape sequence to abort.\nSending 5, 100-byte ICMP
Echos to 10.10.10.254, timeout is 2 seconds:\n!!!!!\nSuccess rate is 100 percent (5/5),
round-trip min/avg/max = 1/2/9 ms"], "stdout_lines": [["Type escape sequence to abort.",
"Sending 5, 100-byte ICMP Echos to 10.10.10.254, timeout is 2 seconds:", "!!!!!",
"Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/9 ms"]]}
ok: [SP1R1] => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python"},
"changed": false, "stdout": ["Type escape sequence to abort.\nSending 5, 100-byte ICMP
Echos to 10.10.10.254, timeout is 2 seconds:\n!!!!!\nSuccess rate is 100 percent (5/5),
round-trip min/avg/max = 1/2/9 ms"], "stdout_lines": [["Type escape sequence to abort.",
"Sending 5, 100-byte ICMP Echos to 10.10.10.254, timeout is 2 seconds:", "!!!!!",
"Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/9 ms"]]}
ok: [SP1R3] => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python"},
"changed": false, "stdout": ["Type escape sequence to abort.\nSending 5, 100-byte ICMP
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
Echos to 10.10.10.254, timeout is 2 seconds:\n!!!!!\nSuccess rate is 100 percent (5/5),
round-trip min/avg/max = 1/2/9 ms"], "stdout_lines": [["Type escape sequence to abort.",
"Sending 5, 100-byte ICMP Echos to 10.10.10.254, timeout is 2 seconds:", "!!!!!",
"Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/9 ms"]]}
ok: [SP1R2] => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python"},
"changed": false, "stdout": ["Type escape sequence to abort.\nSending 5, 100-byte ICMP
Echos to 10.10.10.254, timeout is 2 seconds:\n!!!!!\nSuccess rate is 100 percent (5/5),
round-trip min/avg/max = 1/1/1 ms"], "stdout_lines": [["Type escape sequence to abort.",
"Sending 5, 100-byte ICMP Echos to 10.10.10.254, timeout is 2 seconds:", "!!!!!",
"Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms"]]}
META: ran handlers
META: ran handlers

PLAY [IOS XE ping management station]
*********************************************************************
META: ran handlers

TASK [ping management station in VRF]
*********************************************************************
task path: /home/nso/pod-ping-management.yml:17
ok: [SP1R5] => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python"},
"changed": false, "stdout": ["Type escape sequence to abort.\nSending 5, 100-byte ICMP
Echos to 10.10.10.254, timeout is 2 seconds:\n!!!!!\nSuccess rate is 100 percent (5/5),
round-trip min/avg/max = 1/1/1 ms"], "stdout_lines": [["Type escape sequence to abort.",
"Sending 5, 100-byte ICMP Echos to 10.10.10.254, timeout is 2 seconds:", "!!!!!",
"Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms"]]}
META: ran handlers
META: ran handlers

PLAY RECAP
*********************************************************************************
SP1R1 : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
SP1R2 : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
SP1R3 : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
SP1R4 : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
SP1R5 : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3: Basic NSO concepts

Task 3.1: Introducing NSO

Open an SSH session to your Automation and Management host.

In our pods, NSO is already preinstalled as a package, but needs some configuration. We'll familiarize ourselves with the starting, stopping and troubleshooting NSO.

NSO is preinstalled in your pod in `nso-$VERSION` directory, for example in `/home/nso/nso-5.3`.

As NSO requires some paths present in your shell, you should start with "sourcing" one of the files – ncsrc or ncsrc.tsch depending on the shell you're using (first is for sh and Bash, the other is for tcsh):

```
nso@nso:~/nso-5.3$ pwd
/home/nso/nso-5.3
nso@nso:~/nso-5.3$ source ncsrc
```

To clone existing install execute from within `nso-$VERSION` directory:

```
# cd /home/nso/nso-5.3
# ncs-setup --dest /home/nso/ncs-run
```

NSO will be cloned to `/home/nso/ncs-run` in this example.



Please be aware, that at any time you may be running multiple instances of NSO if configured to use different ports (otherwise, there may be conflicts with binding default ports).

Because of that, it's important to understand how to stop and/or restart NSO. Generally speaking, because **ncs** series tools are in your path, you should be able to execute ncs commands everywhere within your directory tree. If you're working with multiple NSO instances, you need to reference specific config file using `` `-c config-file-name.conf` `` for ncs command to understand which specific instance to operate with.

In our lab, we're using one instance.

You may check if that was executed properly by moving to that directory and then checking the directory tree:

```
nso@nso:~/nso-5.3$ cd ../ncs-run/
nso@nso:~/ncs-run$ ls
README.ncs  logs  ncs-cdb  ncs.conf  packages  scripts  state
nso@nso:~/ncs-run$ tree
.
├── README.ncs
├── logs
├── ncs-cdb
├── ncs.conf
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
├── packages
├── scripts
│   ├── command
│   └── post-commit
└── state

7 directories, 2 files
```

To start NSO, you either execute `ncs` command with specific `ncs.conf` provided with `-c config-file-name.conf` switch, or simply `ncs` in the cloned NSO directory (in which case, `ncs.conf` file will be read from current directory).

Let's start our instance of NSO. As we're already in `/home/nso/ncs-run` cloned directory, we can simply execute:

```
nso@nso:~/ncs-run$ ncs -c ./ncs.conf
```

To check NSO status, you'll do:

```
nso@nso:~/ncs-run$ ncs --status | grep status:
status: started
```

If you get output similar to what's below:

```
nso@nso:~/ncs-run$ ncs -c ./ncs.conf
Cannot bind to internal socket 127.0.0.1:4569 : address already in use
Daemon died status=20
```

It means that likely already one instance of NSO is running and you just tried to run second one. That's possible, but requires modifying ncs.conf file and is outside of scope of this bootcamp.

As in default configuration, NSO listens on port 8080 among others, it's easy to check if that's indeed NSO taking the port (`ncs.smp` is process name for NSO):

```
nso@nso:~/ncs-run$ netstat -nap | grep 8080
tcp        0      0 0.0.0.0:8080            0.0.0.0:*               LISTEN      23925/ncs.smp
```

Obviously, to stop NSO instance you'd then do:

```
[ncs-run]$ ncs --stop
```

If everything went correctly and no NSO process is running anymore, any `ncs` command will report `connection refused` (as the control command cannot communicate with the server):

```
[ncs-run]$ ncs --status | grep status
connection refused (status)
```

If you need to check logs for any troubleshooting information, NSO stores most of the information in **ncs.log** file that's located in **ncs-run/logs/** directory (along with **audit.log** and **devel.log** and others).

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2: Staring your NSO instance

Given we just cloned NSO instance to `/home/nso/nso-5.3` and then experimented with starting it up and then shutting it down, to work with NSO we have to make sure instance is up.

Please execute to bring NSO instance once again up:

```
nso@nso:~/ncs-run$ ncs -c ./ncs.conf
```

NSO interaction can be done in multiple ways. Common are:

- NSO CLI, entered from Linux shell using command ncs_cli; actually, it's a daemon that listens on 2024/tcp socket for SSH connections
- NSO HTTP interface, provided by default configuration and running on port 8080/tcp; this gives also API (REST/RestConf) access if needed
- NSO HTTPS interface, disabled by default, and configured to run on port 8888/tcp



If for some reason in the future you'll need to adjust that, everything NSO does is controlled from main configuration file ncs.conf.

For SSH and Web interfaces (including default API access point), here are relevant ncs.conf file sections:

```
<cli>
    <enabled>true</enabled>

    <!-- Use the builtin SSH server -->
    <ssh>
      <enabled>true</enabled>
      <ip>0.0.0.0</ip>
      <port>2024</port>
    </ssh>

<webui>
    <enabled>true</enabled>
    <transport>
      <tcp>
        <enabled>true</enabled>
        <ip>0.0.0.0</ip>
        <port>8080</port>
      </tcp>
      <ssl>
        <enabled>false</enabled>
        <ip>0.0.0.0</ip>
        <port>8888</port>
     </ssl>
    </transport>
   </cgi>
  </webui>

  <rest>
    <enabled>true</enabled>
  </rest>

  <restconf>
    <enabled>true</enabled>
  </restconf>
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

In other means, NSO will by default listen on port 2024/tcp for SSH “console” access, and on 8080/tcp for WEB/API access.

Task 3.2.1: Adding NEDs to our NSO instance

As NSO needs NEDs (Network Element Drivers) to operate on any devices (IOS-XE and IOS XR routers in our lab), we’ll need to copy relevant NEDs to our cloned instance before we can start configuring anything.



NSO comes with some default NED modules, and further can be purcharsed.

Of course, you can also write NEDs yourself.

As you can see, working copy of our NSO instance doesn’t have any NEDs, so our NSO instance won’t be able to connect to any devices:

```
nso@nso:~/ncs-run$ cd packages/
nso@nso:~/ncs-run/packages$ ls
```

At that same time, NSO master installation has couple of different Cisco related NEDs:

```
nso@nso:~/ncs-run/packages$ ls -d /home/nso/nso-5.3/packages/neds/cisco-*
/home/nso/nso-5.3/packages/neds/cisco-asa-cli-6.6
/home/nso/nso-5.3/packages/neds/cisco-ios-cli-3.0
/home/nso/nso-5.3/packages/neds/cisco-ios-cli-3.8
/home/nso/nso-5.3/packages/neds/cisco-iosxr-cli-3.0
/home/nso/nso-5.3/packages/neds/cisco-iosxr-cli-3.5
/home/nso/nso-5.3/packages/neds/cisco-nx-cli-3.0
```

We will copy NED directories from original /home/nso/nso-5.3/packages/neds/ folder, to our cloned /home/nso/ncs-run/packages/neds/ by executing:

```
nso@nso:~/ncs-run/packages$ cp -R /home/nso/nso-5.3/packages/neds/cisco-ios-cli-3.8 .
nso@nso:~/ncs-run/packages$ cp -R /home/nso/nso-5.3/packages/neds/cisco-iosxr-cli-3.5 .
nso@nso:~/ncs-run/packages$ ls -la /home/nso/ncs-run/packages/
total 16
drwxrwxr-x 4 nso nso 4096 Jun  6 11:31 .
drwxrwxr-x 8 nso nso 4096 Jun  6 11:19 ..
drwxr-xr-x 7 nso nso 4096 Jun  6 11:31 cisco-ios-cli-3.8
drwxr-xr-x 8 nso nso 4096 Jun  6 11:31 cisco-iosxr-cli-3.5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.2.2: Working with NSO CLI

Now, we're going to login to NSO CLI interface and request rebuilding of packages used to connect to different devices:

```
nso@nso:~/ncs-run/packages$ ncs_cli -u admin
admin connected from 192.168.13.1 using ssh on nso
admin@ncs>
```

By default (that can be customized), NSO uses JunOS-alike CLI. To switch to Cisco-like CLI execute following command:

```
admin@ncs> switch cli
```

Now, request reloading (refreshing) NEDs that are available to currently running NSO instance:

```
admin@ncs# packages reload

>>> System upgrade is starting.
>>> Sessions in configure mode must exit to operational mode.
>>> No configuration changes can be performed until upgrade has completed.
>>> System upgrade has completed successfully.
reload-result {
    package cisco-ios-cli-3.8
    result true
}
reload-result {
    package cisco-iosxr-cli-3.5
    result true
}
admin@ncs#
System message at 2021-01-06 11:33:26...
    Subsystem started: ncs-dp-1-cisco-ios-cli-3.8:IOSDp2
admin@ncs#
System message at 2021-01-06 11:33:26...
    Subsystem started: ncs-dp-2-cisco-ios-cli-3.8:IOSDp
```

After the NEDs were refreshed, we can now configure access to our Nexus switches. First, we need to store auth credentials for use by NSO:

```
admin@ncs# conf
Entering configuration mode terminal
admin@ncs(config)# devices authgroups group CCIESP5 default-map remote-name cisco
remote-password cisco
admin@ncs(config-group-CCIESP5)# commit
Commit complete.
```

Notice, NSO also has ‘commit’ concept, just like IOS XR. That will help while working in IOS-XE sessions or with any other system that doesn’t have 2-level commit feature.

Once this is done, NSO can successfully authenticate to devices within this authentication group. Let's add devices we'll use in this lab:

```
admin@ncs(config)# devices device SP1R1 address 10.10.10.11 authgroup CCIESP5
admin@ncs(config-device-SP1R1)# device-type cli ned-id cisco-iosxr-cli-3.5
admin@ncs(config-device-SP1R1)# state admin-state unlocked
admin@ncs(config)# devices device SP1R2 address 10.10.10.12 authgroup CCIESP5
admin@ncs(config-device-SP1R2)# state admin-state unlocked
admin@ncs(config-device-SP1R2)# device-type cli ned-id cisco-iosxr-cli-3.5
admin@ncs(config)# devices device SP1R3 address 10.10.10.13 authgroup CCIESP5
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
admin@ncs(config-device-SP1R3)# device-type cli ned-id cisco-iosxr-cli-3.5
admin@ncs(config-device-SP1R3)# state admin-state unlocked
admin@ncs(config)# devices device SP1R4 address 10.10.10.14 authgroup CCIESP5
admin@ncs(config-device-SP1R4)# state admin-state unlocked
admin@ncs(config-device-SP1R4)# device-type cli ned-id cisco-iosxr-cli-3.5
admin@ncs(config)# devices device SP1R5 address 10.10.10.15 authgroup CCIESP5
admin@ncs(config-device-SP1R5)# device-type cli ned-id cisco-ios-cli-3.8
admin@ncs(config-device-SP1R5)# state admin-state unlocked
admin@ncs(config-device-SP1R5)# commit
Commit complete.
```

Now we have to get SSH host keys, otherwise NSO will not (by default) be willing to connect to them:

```
admin@ncs# devices fetch-ssh-host-keys
fetch-result {
    device SP1R1
    result updated
    fingerprint {
        algorithm ssh-rsa
        value d4:9a:66:da:27:b3:20:1c:a7:c9:88:75:93:6a:6d:aa
    }
}
fetch-result {
    device SP1R2
    result updated
    fingerprint {
        algorithm ssh-rsa
        value d2:0d:da:af:30:6a:92:72:57:40:9a:eb:6b:b6:36:f3
    }
}
fetch-result {
    device SP1R3
    result updated
    fingerprint {
        algorithm ssh-rsa
        value e0:14:cc:1c:7f:f6:aa:f3:9a:d9:0e:51:98:82:94:8e
    }
}
fetch-result {
    device SP1R4
    result updated
    fingerprint {
        algorithm ssh-rsa
        value d0:c6:1c:85:ad:92:49:ed:6b:f0:a3:01:f3:51:cc:22
    }
}
fetch-result {
    device SP1R5
    result updated
    fingerprint {
        algorithm ssh-rsa
        value 67:d7:ce:cd:4b:c2:24:f9:1d:fc:d3:5a:4b:6e:fa:39
    }
}
```

Having done so, we can now request NSO to refresh its database with device configs:

```
admin@ncs# devices sync-from
sync-result {
    device SP1R1
    result true
}
sync-result {
    device SP1R2
    result true
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
}
sync-result {
    device SP1R3
    result true
}
sync-result {
    device SP1R4
    result true
}
sync-result {
    device SP1R5
    result true
}
```

Once we have this data in DB, we can request different operations. Easiest one is to just see the config modularized by NSO feature understanding:

```
admin@ncs# show running-config devices device SP1R4 config
devices device SP1R4
 config
  hostname  SP1R4
  vrf Mgmt-intf
   address-family ipv4 unicast
   exit
   address-family ipv6 unicast
   exit
  exit
  vrf sitec
[…]
```

As you see, the config per-device is modularized. NSO will enable us to work with this data. The data can be represented in different output formats.

You can also use NED ability to work with logical modules on OS side (IOS XR in this example) to show you only specific area of configuration. For example, let's just see interfaces from SP1R4 configuration:

```
admin@ncs# show running-config devices device SP1R4 config interface
devices device SP1R4
 config
  interface Loopback 0
   description Loopback
   ipv4 address 33.56.4.4 255.255.255.255
   ipv6 address 2001:33:56:aa::4/128
  exit
  interface Loopback 1
   vrf sitec
   ipv4 address 172.16.4.41 255.255.255.255
   ipv6 address 2001:33:56::4:41/128
  exit
[…]
```

We can also request specific display output options, for example ask for JSON or XML format. Let's use XML as it will be handy later:

```
admin@ncs# show running-config devices device SP1R4 config interface | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <devices xmlns="http://tail-f.com/ns/ncs">
  <device>
    <name>SP1R4</name>
      <config>
      <interface xmlns="http://tail-f.com/ned/cisco-ios-xr">
        <Loopback>
          <id>0</id>
          <description>Loopback</description>
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

```
          <ipv4>
            <address>
              <ip>33.56.4.4</ip>
              <mask>255.255.255.255</mask>
            </address>
[…]
```

...and here the same output but in JSON format:

```
admin@ncs# show running-config devices device SP1R4 config interface | display json
{
  "data": {
    "tailf-ncs:devices": {
      "device": [
        {
          "name": "SP1R4",
          "config": {
            "tailf-ned-cisco-ios-xr:interface": {
              "Loopback": [
                {
                  "id": 0,
                  "description": "Loopback",
                  "ipv4": {
                    "address": {
                      "ip": "33.56.4.4",
                      "mask": "255.255.255.255"
                    }
                  },
[…]
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 3.3: Saving and loading device information

While working with specific configuration in NSO, you may be tempted to save specific configuration parts into file for importing later.

To save device configuration to ASCII file, execute ‘save’ in configuration mode:

```
admin@ncs# conf
Entering configuration mode terminal
admin@ncs(config)# save ?
Possible completions:
  <filename>
admin@ncs(config)# save config.txt
```

In future, if you want to load something from file (let’s say AAA information), you can load the text file back. While doing so, you have three options how the existing configuration will be treated vs loaded content:

```
admin@ncs(config)# load ?
Possible completions:
  merge    - Merge with existing configuration
  override - Overwrite old configuration
  replace  - Replace old configuration
admin@ncs(config)# load merge ?
Possible completions:
  <filename>  config.txt
```

If you’re storing snippets of configuration in different files, loading ‘merge’ will make most sense. If you just want to overwrite existing NSO configuration – choose ‘replace’.

© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

Task 4: Advanced NSO concepts

Task 4.1: Cisco NSO template basics

Next exercise is to use simple configuration template (we'll use fictious NTP configuration example) and ask NSO to compile us compliance report.

This is handy, when you're using NSO to periodically report on configuration discrepancies.

Our server configuration will be common – routers should synchronize time with server located at 10.10.10.1.

To define template from NSO CLI, let's go through this simple configuration:

```
admin@ncs(config)# devices template CCIESP5-NTP
admin@ncs(config-template-CCIESP5-NTP)# ned-id cisco-iosxr-cli-3.5
admin@ncs(config-ned-id-cisco-iosxr-cli-3.5)# config
admin@ncs(config-config)# ntp server server-list 10.10.10.1
admin@ncs(config-server-list-10.10.10.1)# exit
admin@ncs(config-config)# exit
admin@ncs(config-ned-id-cisco-iosxr-cli-3.5)# exit
admin@ncs(config-template-CCIESP5-NTP)# exit
admin@ncs(config)# exit
Uncommitted changes found, commit them? [yes/no/CANCEL] yes
Commit complete.
```

You can verify how the template looks in XML form and even export it to file for further reuse:

```
admin@ncs(config)# show full-configuration devices template CCIESP5-NTP | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <devices xmlns="http://tail-f.com/ns/ncs">
  <template>
    <name>CCIESP5-NTP</name>
    <ned-id>
      <id xmlns:cisco-iosxr-cli-3.5="http://tail-f.com/ns/ned-id/cisco-iosxr-cli-
3.5">cisco-iosxr-cli-3.5:cisco-iosxr-cli-3.5</id>
      <config>
        <ntp xmlns="http://tail-f.com/ned/cisco-ios-xr">
          <server>
            <server-list>
              <name>10.10.10.1</name>
            </server-list>
          </server>
        </ntp>
      </config>
    </ned-id>
  </template>
  </devices>
</config>
```

Now, let's apply freshly defined NTP template:

```
admin@ncs(config)# devices device SP1R1 apply-template template-name CCIESP5-NTP
apply-template-result {
    device SP1R1
    result ok
}
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński

We can check how the configuration will look like on router:

```
admin@ncs(config)# show configuration
devices device SP1R1
 config
  ntp
   server 10.10.10.1
  exit
 !
!
```

…and commit it:

```
admin@ncs(config)# commit
Commit complete.
```

When you log in back to SP1R1, you should see new configuration done:

```
RP/0/0/CPU0:SP1R1#sh running-config ntp
ntp
 server 10.10.10.1
!
```


© 2015-2021 Łukasz Bromirski & Piotr Jabłoński