

CCDE Quick Reference

Russ White, CCIE No.

Mosaddaq Turabi, CCIE No. 1864

Chapter 1: The Cisco Certified Design Expert

Chapter 2: EIGRP Theory and Operation

Chapter 3: OSPF Theory and Operation

Chapter 4: IS-IS Theory and Operation

Chapter 5: BGP Theory and Operation

Chapter 6: General Routed Design Theory

Chapter 7: Topology Design Theory

Chapter 8: Tunneling Technologies

Chapter 9: Network Management

Chapter 10: IP Quality of Service



Your Short Cut to Knowledge

ciscopress.com



CCDE Quick Reference

Russ White, CCIE No.

Mosaddaq Turabi, CCIE No. 1864

Chapter 1: The Cisco Certified Design Expert

Chapter 2: EIGRP Theory and Operation

Chapter 3: OSPF Theory and Operation

Chapter 4: IS-IS Theory and Operation

Chapter 5: BGP Theory and Operation

Chapter 6: General Routed Design Theory

Chapter 7: Topology Design Theory

Chapter 8: Tunneling Technologies

Chapter 9: Network Management

Chapter 10: IP Quality of Service



Your Short Cut to Knowledge

ciscopress.com

CCDE Quick Reference

Russ White
Mosaddaq Turabi



ciscopress.com

CCDE Quick Reference

Russ White and Mosaddaq Turabi

Copyright © 2009 Cisco Systems, Inc.

Published by:
Cisco Press
800 East 96th Street
Indianapolis, Indiana 46240 USA

All rights reserved. No part of this digital short cut may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

First Digital Edition April 2009
ISBN-10: 1-58705-839-1
ISBN-13: 978-1-58705-839-4

Warning and Disclaimer

This digital Short Cut is designed to provide information about networking. Every effort has been made to make this digital Short Cut as complete and accurate as possible, but no warranty or fitness is implied. The information is provided on an "as is" basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this digital short cut.

The opinions expressed in this digital Short Cut belong to the authors and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this digital Short Cut that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc. cannot attest to the accuracy of this information. Use of a term in this digital Short Cut should not be regarded as affecting the validity of any trademark or service mark.

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members of the professional technical community.

Reader feedback is a natural continuation of this process. If you have any comments on how we could improve the quality of this digital short cut, or otherwise alter it to better suit your needs, you can contact us through e-mail at feedback@ciscopress.com. Please be sure to include the digital Short Cut title and ISBN in your message.

We greatly appreciate your assistance.

Corporate and Government Sales

The publisher offers excellent discounts on this digital short cut when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact: **U.S. Corporate and Government Sales** 1-800-382-3419 corpsales@pearsontechgroup.com. For salesc outside the United States please contact: **International Sales** international@pearsoned.com

---



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

♻ CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS,

Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.
All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

As a final exam-preparation tool, the CCDE Quick Reference provides a concise review of all objectives on the CCDE written exam (351-001). This Digital Short Cut provides you with detailed, graphical-based information, highlighting only the key topics in cram-style format.
With this document as your guide, you will review topics on network design in the areas of routing, tunneling, quality of service, management, cost, capacity, and security. This fact-filled Quick Reference allows you to get all-important information at a glance, helping you focus your study on areas of weakness and enhancing memory retention of essential exam concepts.

Contents

Chapter 1: The Cisco Certified Design Expert
Chapter 2: EIGRP Theory and Operation
Chapter 3: OSPF Theory and Operation
Chapter 4: IS-IS Theory and Operation
Chapter 5: BGP Theory and Operation
Chapter 6: General Routed Design Theory
Chapter 7: Topology Design Theory
Chapter 8: Tunneling Technologies
Chapter 9: Network Management
Chapter 10: IP Quality of Service

About the Authors

Russ White, CCIE No. 2635, is a member of the Routing Protocol Design and Architecture Team at Cisco, Research Triangle Park, North Carolina. He is a member of the IETF Routing Area Directorate, co-chair of the Routing Protocols Security Working Group in the IETF, a regular speaker at Cisco Live, a member of the CCIE Content Advisory Group, a member of the core team developing the new Cisco Design certification, a regular contributor to the *Internet Protocol Journal*, and the co-author of six other books about routing and routing protocols, including *Optimal Routing Design*, from Cisco Press. Russ primarily works in the development of new features and design architectures for routing protocols.

Mosaddaq Turabi, CCIE 1864, is a Distinguished Support Engineer at Cisco Systems. After earning a bachelor of science degree and a master of science degree in electrical engineering with a focus on digital design, Mosaddaq joined Cisco in 1995 to work with Cisco's TAC team. Subsequently, Mosaddaq moved to the ISP Support team, where he supported large service providers in the areas of network architecture, design, and deployment. Mosaddaq has been involved in the design, implementation, and planning of numerous large-scale IP and MPLS networks. He has provided consulting services for the introduction and deployment of MPLS and QoS-based services within service provider and large-scale enterprise networks.

About the Technical Editor

John Cavanaugh is a Distinguished Services Engineer in the Central Engineering (CE) Architecture and Design, Advanced Services organization at Cisco. He is a renowned expert in the architecture and designs of high-availability network infrastructure and data centers, and has worked on projects for most of the global financial institutions.

Chapter 1. The Cisco Certified Design Expert

The Cisco Certified Design Expert (CCDE) is an expert-level certification providing both an indicator of skill level in network design theory and application and a target for network engineers wanting to learn and apply network design principles. The CCDE is designed for engineers with at least seven years of experience in network design and operations.
The CCDE covers a wide range of technologies and concepts, including the following:

- **Routing protocol theory and operation**: To obtain the CCDE, a candidate must know and understand the theory of operation for each of the four primary routing protocols in widespread use today: Open Shortest Path First (OSPF) Protocol, Enhanced Interior Gateway Protocol (EIGRP), the Intermediate System-to-Intermediate System (IS-IS) Protocol, and the Border Gateway Protocol (BGP) Version 4.

- **Routing protocol application and design**: To obtain the CCDE, a candidate must understand the application of routing protocol design to various business, service provisioning, and technical design problems. The candidate must understand how routing design impacts application performance, network resiliency, and operational expenses.

- **Tunneling theory and design**: To obtain the CCDE, a candidate must understand the operation of various tunneling technologies, what the characteristics of each tunnel type is, how each tunneling technology impacts business and application requirements, and the fundamental principles employed in designing their deployment.

- **Quality of service**: To obtain the CCDE, a candidate must understand each type of quality of service available, and how each can be applied to business and application requirements.

- **Network management**: To obtain the CCDE, a candidate must understand the principles of well-designed network management. Network management, to a good designer, is not an afterthought, but is rather built in to the design and configuration of the network as a whole.

- **Security**: To obtain the CCDE, a candidate must understand a broad array of security concepts, including providing resiliency during attacks, defense in depth, blocking attacks at an early point, and the tools available to use and protect the network infrastructure.

Each of these concepts is tested on both the written prequalification test and on the CCDE practical examination. The following chapters provide an outline or overview of the information covered on the CCDE written prequalification test in each of these areas.

The CCDE practical exam is not covered in this Quick Reference Guide.

When considering each of the following chapters, remember that all the material is to be approached from a network design perspective. In essence, for any piece of information, ask this simple question: "Would knowing this piece of information change the way I design a network in any substantial way?" If the answer is no, the information is not likely to be tested on the CCDE.

Chapter 2. EIGRP Theory and Operation

Enhanced Interior Gateway Protocol (EIGRP) is a widely used advanced distance vector routing protocol. This chapter provides a high-level overview of the theory and operation of EIGRP.

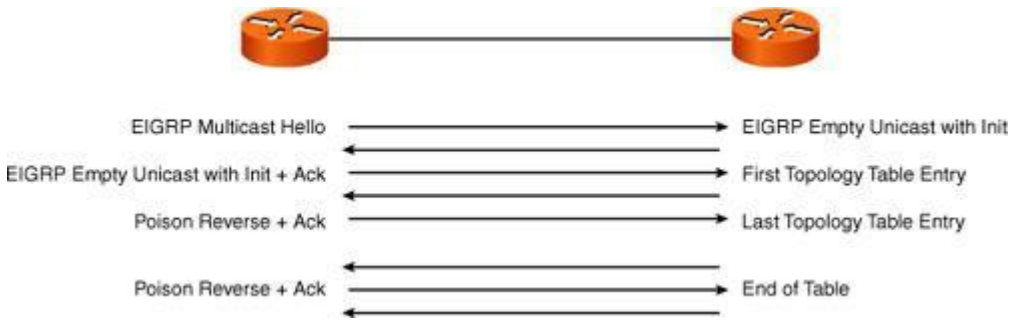
Neighbor Relationships

EIGRP forms neighbor relationships for two primary reasons:

- To maintain state about reachable destinations and paths without periodically retransmitting this information
- To provide a reliable mechanism to process the loss of reachability information

EIGRP uses a three-way handshake process to build a neighbor adjacency, as shown in Figure 2-1.

FIGURE 2-1 EIGRP Neighbor Formation



Once EIGRP neighbors are formed, they are maintained through the use of periodic hello messages between the neighbors. If no other EIGRP packet is transmitted during the hello interval, EIGRP transmits a hello

to maintain the neighbor relationship. If an EIGRP router does not hear from a neighbor within the hold timer, it resets the neighbor relationship. The primary points of interest in the formation and maintenance of EIGRP neighbors are as follows:

- EIGRP will not form neighbor relationships unless both unicast and multicast packets are being transmitted and received across the link.

- EIGRP will not form neighbor adjacencies across a link with mismatched IP addresses.

- EIGRP paces the rate at which it transmits packets on multipoint links configured with less than 1500 kb/s of bandwidth. The bandwidth configured on the interface is used to calculate the rate at which packets are transmitted, or paced.

- EIGRP defaults to a 60-second hello timer and a 180-second hold timer on multipoint links configured with a bandwidth of less than 1500 kbps. On all other links, EIGRP defaults to a 5-second hello timer and a 15-second hold timer.

- EIGRP can be configured to transmit a hello as quickly as once every second, with a corresponding 3-second hold timer.

- The EIGRP hold timer used for any given neighbor is the hold timer advertised by that neighbor. This means the EIGRP hold timers for a given link do not need to match, because each neighbor sets its own hold timer based on the frequency at which it expects to transmit hello packets.

EIGRP Metrics

EIGRP relies on a compound metric to determine the cost of a given path to a destination. Five component metrics are carried in EIGRP routing

updates:

- Minimum bandwidth
- Sum of the delays
- Highest link load
- Lowest reliability
- Lowest maximum transmission unit (MTU)

The combination of these metrics is determined by the setting of the *K values*, which are manually configured on each router in the network. By default, the K values are set so the highest link load, lowest reliability, and lowest MTU are ignored, so that the lowest bandwidth and the sum of the delays are the only metrics combined to produce the composite metric. Figure 2-2 shows the formula used to combine these two vector component metrics into a single composite metric.

FIGURE 2-2 EIGRP Metric Calculation

$$\left[\frac{10^7}{\min(bandwidth)} + \sum delays\right] * 256$$

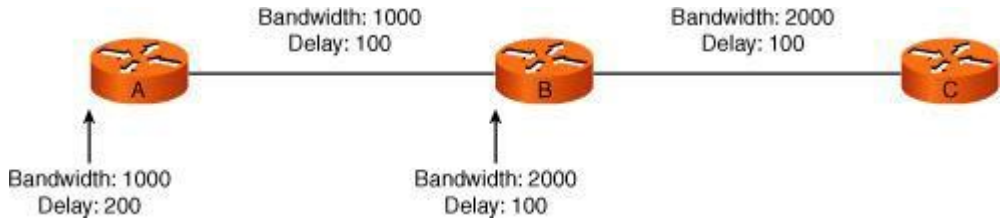
Although this metric formula appears to be complex, it is generally, in practice, simple. You should remember the following:

- If the sum of the delays along every possible path is the same, the path with the lowest bandwidth will always be chosen as the best path.
- If the lowest bandwidth along every possible path is the same, the path with the lowest sum of the delays will always be chosen as the best path.
- The component metrics are only read from the local interfaces when neighbor relationships are established in current versions of Cisco IOS Software. In some Cisco IOS Software versions,

these metrics are read from the interfaces dynamically, allowing for dynamic path selection based on current path status.

EIGRP metrics are added on the inbound interface, as is common with all routing protocols, as shown in Figure 2-3.

FIGURE 2-3 EIGRP Metric Origins



When Router B transmits the update to Router A, it transmits the update with the existing, or local, metrics. When Router A receives the update, it uses the metrics on the inbound interface to modify the metrics in the received update.

EIGRP differentiates between internal and external routing information. A route that is learned from a source external to EIGRP carries the original route metric, route tags, and other information about the originating routing domain. EIGRP always prefers internally learned routing information over information learned from external routing sources.

Some later versions of EIGRP also carry communities, which are a form of route tags, on internal routes. You can set the administrative distance so that EIGRP prefers external routes over internal routes, but this is a dangerous practice, and therefore generally not recommended.

Path Selection

An EIGRP process transmits all routes that meet the following criteria to a neighboring router:

- **Are currently installed in the local routing table by the EIGRP process**: Routes installed by other processes (even other EIGRP processes) are not transmitted to neighbors.

- **Do not use the interface on which the neighbor resides as their outbound interface**: This is the split-horizon rule.

All routes received by an EIGRP router are placed in a local topology table. Once the local topology table is built, routes are chosen based on the following process:

- The route to any given destination with the lowest cost is chosen as the best path. This route is called the *successor*, and the metric of the lowest-cost route is called the *feasible distance*.

- If the local topology table contains multiple equal-cost paths, multiple routes will be installed in the local routing table, up to the limit imposed by the maximum paths configured on the router.

- Any route for which the neighbor's metric (the reported distance) is less than the local feasible distance (the cost of the lowest-cost route) will be marked as a loop-free alternate path, a *feasible successor*.

EIGRP can install both successors and feasible successors in the local routing table, causing traffic to be forwarded across unequal-cost paths. This is controlled by the *variance* command on the local router. Traffic is shared based on the relationship of the EIGRP metrics; the highest-cost path over which traffic will be transmitted is divided by the metric of each remaining path. The resulting number is the proportion of traffic that will be transmitted over that path.

Active Processing

When an EIGRP router loses its best path to a given destination, it follows a simple process to determine an alternate route:

- It examines the local topology table for a route with equal cost. If there are other equal-cost routes, the router shifts traffic to these links.

- If there are no routes with equal costs, the router examines the topology table for feasible successors. If there is a feasible successor, it installs this route into the local routing table, shifting traffic to this path.

- If there are no feasible successors, the router marks the route active and sends a query for this destination, with the last known feasible distance, to all neighbors except those on the same interface as the past successor.

An EIGRP router receiving a query examines its local topology table and determines whether it has any information about the destination in question:

- If it does not have any information about the destination in the query, it transmits a negative reply, including an infinite metric.

- If it does have a topology table entry for this destination, and the query was received from the local successor, it marks the route active and transmits a query to each of its neighbors.

- If it does have a topology table entry for this destination, and the query was not received from the local successor, it replies with the metric of its local successor.

When all of a router's neighbors have replied, it replies to any queries about this destination that it has received and not replied to, recalculates the feasible distance, installs the correct routes in the local routing table, and then transmits a following update so that its neighbors now know which route it is using.

Query Bounding

EIGRP queries are bounded in four ways:

- When a router receives the query from a nonsuccessor neighbor for that route, it replies with the local feasible distance.

- When a router receives a query for which it has no local topology table entry, it replies with an infinite metric. This could be the result of route filters or aggregation.

- When a router receives a query, and has no neighboring routers to query, it replies with an infinite metric.

- Queries are not transmitted to neighboring routers marked as EIGRP stubs.

From a network designer's perspective, EIGRP query boundaries effectively mark failure domain boundaries within the routing system.

Aggregation of Routing Information

A network designer must keep in mind several points about EIGRP aggregation:

- EIGRP aggregates topology information at each hop, by transmitting only the best path among all available paths to each neighbor. EIGRP does have a sense of a two-hop topology, but it is a limited view.

- EIGRP allows aggregation at any point in the network, and in any direction.

- EIGRP installs a discard route that discards traffic to any destination within the aggregate for which no more-specific route exists.

- EIGRP will "autosummarize" along major network boundaries. This doesn't, normally, play a large role in network design any longer because most EIGRP routers are configured to disable this feature.

Chapter 3. OSPF Theory and Operation

Open Shortest Path First (OSPF) Protocol is a widely used advanced link-state routing protocol. This chapter provides a high-level overview of the theory and operation of OSPF.

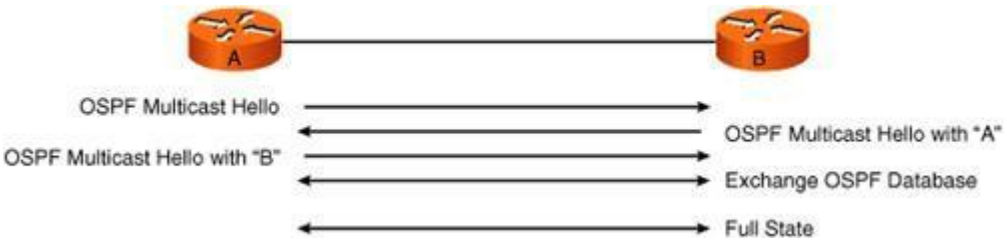
Neighbor Relationships

OSPF forms neighbor relationships for two primary reasons:

- To discover links along which traffic may be forwarded in the network (edges in the shortest-path tree)
- To provide for the reliable transmission of routing information through the network

OSPF uses a three-way handshake process to form neighbor relationships, as shown in Figure 3-1.

FIGURE 3-1 OSPF Neighbor Formation



OSPF will not form a neighbor relationship on an interface if the following parameters do not match:

- The OSPF area ID
- The OSPF area type

- The link maximum transmission unit (MTU)
- The Hello interval
- The dead interval
- The OSPF link type (point to point, broadcast, and so on)

When OSPF neighbors are formed, they are maintained through periodic Hello packets. If a Hello packet is not received from a specific OSPF neighbor at least once each dead interval, the OSPF neighbor is reset. The OSPF Hello timer is set to ten seconds by default, and the dead interval set to 30 seconds by default. On some implementations, the OSPF dead interval can be set to one second or less, with correspondingly short dead intervals.

To reduce flooding cost, and the cost of running the shortest path first (SPF) algorithm, OSPF elects a designated router (DR) on each broadcast link. The DR serves two purposes:

- Link-state updates (link-state acknowledgements, or LSAs), which contain reachable destinations, connections, and other network information, are transmitted to the DR. The DR then refloods this information over the link, which reduces the number of packets flooded and the number of acknowledgment packets across the link.
- The DR creates and maintains a pseudonode for the broadcast link. All the routers on the link advertise a connection to the pseudonode, rather than to each of the OSPF routers connected to the link, which reduces the complexity of the computed shortest-path tree.

The operation of the pseudonode is explained more fully in the “Path Selection” section.

DRs are elected based on two factors:

- The OSPF priority configured on the interface
- The OSPF router ID

The router with the highest OSPF priority will be chosen as the DR. If

all the routers on the link have the same OSPF priority, the router with the highest router ID is elected.
There is also a backup designated router (BDR), which maintains the same state as the DR. This provides for a quick and error-free switchover if the DR fails.
The DR is elected using the following process:

- Each router waits its locally configured dead interval to see whether other routers are already configured on the link and have already elected a DR and BDR.

- Each router connected to the link transmits an OSPF Hello containing its priority, router ID, and the current designated and backup designated routers.

- If there is no current BDR, each router examines the Hellos it has received from each OSPF neighbor and chooses the BDR based on the criteria previously described.

- If there is no current DR, the BDR is promoted to be the DR.

- If there is no BDR, the router selects one based on the criteria previously described.

- Each router now transmits its selections in its Hello packets, and forms neighbor adjacencies on the link.

If any router sends a Hello with an existing DR and BDR onto the link, the remaining routers will select the existing DR and BDR rather than elect a new one. This means the current DR and BDR will remain in place even if a router with a higher OSPF interface priority or higher router ID is attached on the link. In practice, this means the first router connected to a link will end up being the DR, and the second will be the BDR, until some event occurs that changes the initial state or unless all the routers on the link are connected at the same time.
Network engineers need to be careful of two aspects of DR and BDR election:

- The DR should be carefully chosen on some link types,

particularly point-to-multipoint links that are configured as OSPF broadcast links. The DR must be directly reachable from all the routers connected to the link.

- If the router chosen as the DR is important, it should always be configured using the OSPF interface priority, rather than the IP address. The router intended to be the DR should also always be connected to the link first, if possible.

In many implementations, a broadcast link can be configured as a point-to-point link in OSPF to prevent the election of a DR on that link. On broadcast links used in a point-to-point fashion, this can reduce the size of the shortest-path tree.

Path Selection

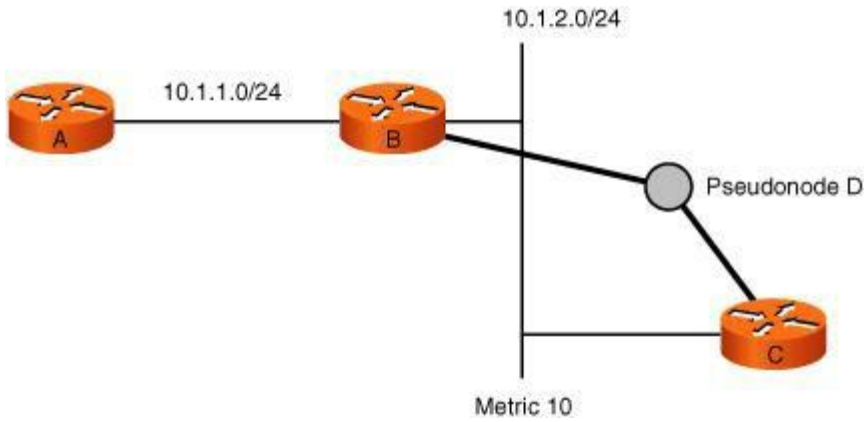
Each OSPF router in the network transmits information about neighbors and links it is connected to in LSAs. Two fundamental types of LSAs are used within a flooding domain (or area) to advertise OSPF information:

- **Router LSAs (Type 1)**: These LSAs contain information about locally connected links and neighbors with which this router has a full neighbor relationship. Each router in the network generates a Router LSA.

- **Network LSAs (Type 2)**: These LSAs advertise pseudonodes in the network. DRs generate Network LSAs.

Figure 3-2 shows the LSAs advertised by an OSPF router.

FIGURE 3-2 OSPF Link-State Advertisements



In Figure 3-2, Router B would advertise the following LSAs:

- **A Router LSA (Type 1) with the following information**
- A connection to Router A with a cost of 10
- A connection to Router D (the pseudonode) with a cost of 10
- A connection to 10.1.1.0/24 with a cost of 10
- A connection to 10.1.2.0/24 with a cost of 10
- **A Network LSA (Type 2) with the following information**
- A connection to Router B with a cost of 0
- A connection to Router C with a cost of 0
- A connection to 10.1.2.0/24

You should notice that the cost from the pseudonode to each connected node is always 0, while the cost to the pseudonode is reported as the link cost on the locally advertising router. This prevents the cost of a single link from being advertised twice, and permits links to have different costs in different directions (asymmetric costs).

OSPF uses Dijkstra's shortest path first (SPF) algorithm to determine the shortest path through the network. Each router transmits a copy of its local link-state database to its fully adjacent neighbors (this process is called *flooding* the database), so each router within a single flooding domain, or area, has the same set of LSAs in its local database.

OSPF runs the SPF algorithm across this database to create a shortest-path tree (SPT). This SPT provides a loop-free path to each destination in the network, and thus provides the basis for the routing information OSPF installs in the local routing table.
OSPF uses a single opaque metric to describe the cost of a link. In many implementations, this metric is chosen based on the configured link bandwidth. The cost each router advertises is the local cost configured on the link it is advertising. It is possible for links to have different costs on each end (asymmetric costs), so that SPF returns different results from different points in the network. This is normally not a problem, as long as no links are 0 cost, other than the links to pseudonodes, as described earlier.
OSPF differentiates between internal and external routing information, too. External routing information is carried in a separate LSA type, an External LSA, which is a Type 5. External routes can have one of two metric types:

- **External Metric Type 1 (E1)**: When calculating the metric to an external destination with an E1 metric, OSPF adds the internal and external metrics to compute a total cost to reach the destination.

- **External Metric Type 2 (E2)**: When calculating the metric to an external destination, OSPF compares the cost to the redistributing router (known as the Autonomous System Border Router, or ASBR). If the E2 metric for any two external routes is equal, the external metrics are compared to determine the best path.

External routes with E1 metrics are always preferred over routes with E2 metrics.
An OSPF router does not generate a new LSA when a link changes immediately, nor does it run SPF immediately on receiving new routing information. Rather, both of the operations have associated timers that help to dampen the pace of change in the network:

- The amount of time an OSPF implementation will wait after noting a link or neighbor change until it transmits an LSA is

variable, but is generally set to around 5 seconds. This timer can normally be adjusted through manual configuration, and can sometimes be set to back off exponentially to prevent a large number of LSAs from being generated quickly.

- The amount of time an OSPF implementation will wait after receiving an LSA before running SPF is variable, but is generally set to around 5 seconds. This timer can normally be adjusted through manual configuration, and can sometimes be set to back off exponentially as LSAs are received more quickly.

Network engineers would normally adjust the LSA generation and SPF timers to improve network convergence times. The tradeoff is as these timers are reduced, the network becomes less stable as the rate of changes increases.

Aggregation of Routing Information

OSPF aggregates topology information by allowing the network engineer to break the network up into multiple flooding domains, which are called *areas*. A router that connects two areas is called an Area Border Router, or ABR. OSPF always has one area 0, which is called the *backbone area*, and a number of outlying areas, which are identified by their area number.

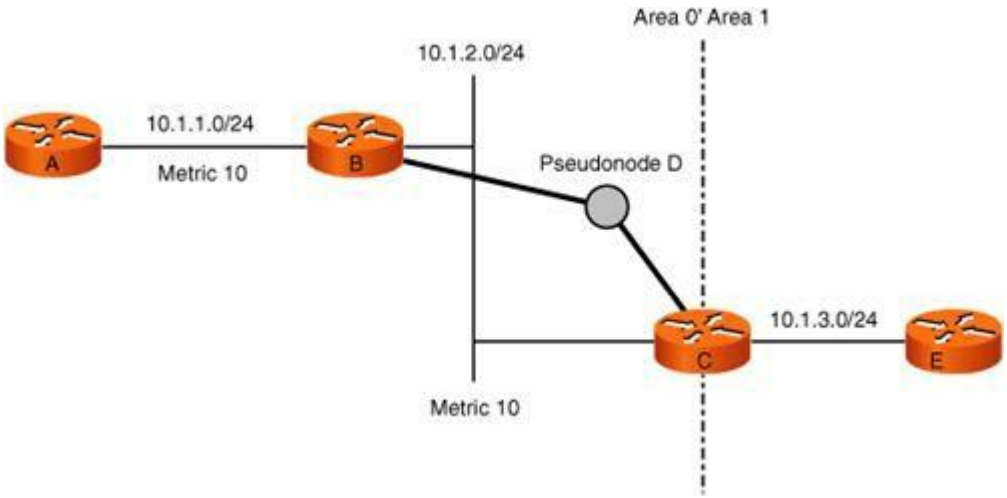
Note

The area numbers in OSPF are not significant, other than area 0. All the areas within an OSPF network can have the same area number. As long as they are not connected to one another, they are still treated as different areas, or flooding domains, by OSPF.

When an ABR receives a Router or Network LSA, it takes the reachable

destinations within the received LSA, and places them into a Summary LSA, which is a Type 3, and advertises the Summary LSA into the "other" area. This makes the destinations appear to be directly connected to the ABR to routers outside the local area, and blocks the flooding of topology information, as shown in Figure 3-3.

FIGURE 3-3 OSPF Area Boundaries



Each of the routers illustrated would have the following LSAs:

- **Routers A, B, and C would all have the following in their area 0 LSA database**
- A Router LSA from Router A containing
- A connection to 10.1.1.0/24
- A connection to B
- A Router LSA from Router B containing
- A connection to 10.1.1.0/24
- A connection to 10.1.2.0/24
- A connection to Router A

- A connection Router D (the pseudonode)
- A Router LSA from Router C containing
- A connection to 10.1.2.0/24
- A connection to Router D (the pseudonode)
- A Network LSA from Router D containing
- A connection to 10.1.2.0/24
- A connection to Router B
- A connection to Router C
- A Summary LSA from Router C containing
- A connection to 10.1.3.0/24
- **Routers C and E would have the following in the area 1 LSA database**
- A Router LSA from Router C containing
- A connection to 10.1.3.0/24
- A connection to Router E
- A Router LSA from Router E containing
- A connection to 10.1.3.0/24
- A connection to Router C
- A Summary LSA from Router C containing:
- A connection to 10.1.1.0/24

• A connection to 10.1.2.0/24

Important points to consider about OSPF areas and ABR operation include the following:

- Routes learned from Router and Network LSAs (intra-area routes) are always preferred over routes that are learned through Summary LSAs (interarea routes).

- External LSAs (Type 5 LSAs) are not included in the Summary LSA generated by the ABR. These are flooded throughout the entire network, except stub areas, as described later.

- The ABR has two OSPF databases, one for each area in which it participates. It treats each of these databases independently, running SPF on each one as changes are required. A change in one area will cause the ABR to run SPF in that area, and then regenerate the Summary LSA it is creating into the other areas it is attached to.

- Aggregation of reachability information and route filtering can be configured only at ABRs. Both of these techniques hide reachability information and topology information within the area toward which the aggregation or filtering is configured.

An OSPF area can be configured in a number of ways, beyond what has been described so far in this chapter. Table 3-1 provides information about each type of OSPF stub area available and its characteristics.

Table 3-1 OSPF Stub Areas and Their Characteristics

Stub Area Type	LSAs Blocked into the Area	External Routes within the Area	Default Route Generation
OSPF stub	Summary LSAs (Type 3s)	Flooded into the area, but cannot be generated in the area	User must ensure correct default route is configured.
OSPF totally stub	Summary LSAs (Type 3s) and External LSAs (Type 5s)	Cannot be generated in the area	Default is injected into the area using a Summary LSA (Type 3) by the ABR.
OSPF not-so-stubby	Summary LSAs (Type 3s)	Flooded into the area, generated within the area as Type 7 LSAs, converted to External LSAs (Type 5s) at the ABR	User must ensure correct default route is configured.
OSPF totally not-so-stubby	Summary LSAs (Type 3s) and External LSAs (Type 5s)	Generated within the area as Type 7 LSAs, converted to External LSAs (Type 5s) at the ABR	Default is injected into the area using a Type 7 LSA (External) by the ABR.

To help remember these area types, consider the following:

- Stub blocks Summary LSAs
- Totally stub blocks Summary and External LSAs
- Not-so-stubby areas can contain ASBRs, which generate external routing information as Type 7 LSAs, which are converted to Type 5 LSAs at the ABR

Combining these three concepts provides all four possible OSPF area types.

Chapter 4. IS-IS Theory and Operation

Intermediate System-to-Intermediate System (IS-IS) Protocol is widely used in Europe, in public utility companies, and in global service provider networks. IS-IS was originally developed to provide routing for the ISO protocol stack, including the most commonly recognized transport protocol within that stack, Connectionless Network Service (CLNS). IS-IS was adapted to use for IP routing by the Internet Engineering Task Force (IETF), which has since added IPv6 routing and, most recently, Layer 2 routing capabilities to the protocol.
In IS-IS, a router is called an intermediate system (IS). Both terms are used throughout this document.

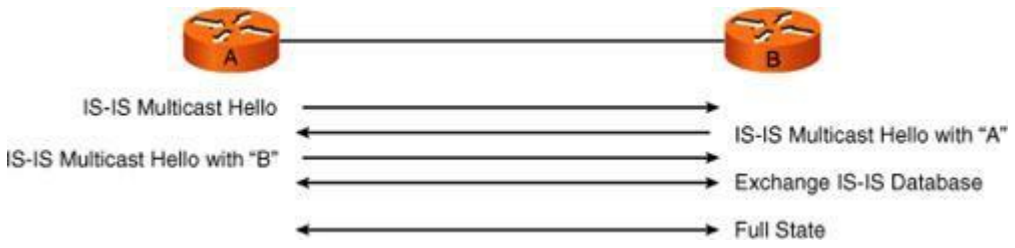
Neighbor Relationships

IS-IS forms neighbor relationships for two primary reasons:

- To discover links along which traffic may be forwarded in the network (edges in the shortest-path tree, SPT)
- To provide for the reliable transmission of routing information through the network

IS-IS uses a three-way handshake process to form neighbor relationships, as shown in Figure 4-1.

FIGURE 4-1 IS-IS Neighbor Formation



IS-IS pads its hello packets to the interface maximum transmission unit (MTU) to make certain packets of that size can be transmitted along the link. This ensures the larger packets containing routing information can traverse the link before the neighbor relationship is formed.

Once IS-IS neighbors are formed, they are maintained through periodic hello packets. If a hello packet is not received from a specific IS-IS neighbor at least once each dead interval, the IS-IS neighbor is reset.

IS-IS treats level 1 routing and level 2 routing (explained further in the section on flooding domains later in this chapter) almost as if they are separate routing processes. Separate hello packets and routing information packets are transmitted for level 1 routing and level 2 routing. The types of routing allowed on the link and the system addresses must match for a neighbor relationship to form. For instance, if two routers are configured to be in different areas (explained in the "Intermediate System Addressing" section), they will not form a neighbor adjacency.

To reduce flooding cost, and the cost of running the shortest path first (SPF) algorithm, IS-IS elects a designated intermediate system (DIS) on each broadcast link. The DIS serves two purposes:

- Periodically transmits a complete sequence number packet (CSNP) onto the link describing its local link-state database. If any router attached to the link has information that is not described in the CSNP, or is missing information described in the CSNP, it will synchronize the missing information with the DIS. This ensures all the routers on the link share a common view of the link-state database.

- The DIS creates and maintains a pseudonode for the broadcast link. All the routers on the link advertise a connection to the pseudonode, rather than to each of the IS-IS routers connected to the link, which reduces the complexity of the computed SPT.

The operation of the pseudonode is explained more fully in the "Path Selection" section. The DIS is elected based on the network entity title (NET), the address of the router. The NET is explained in the "Intermediate System Addressing" section.

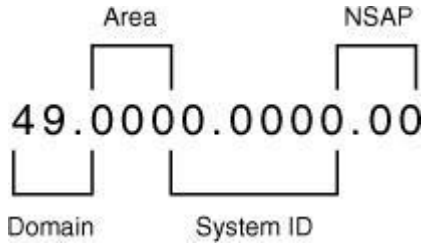
In many implementations, a broadcast link can be configured as a point-

to-point link in IS-IS to prevent the creation of a pseudonode for that link. Configuring broadcast links as point-to-point links can reduce the size of the SPT.

Intermediate System Addressing

IS-IS does not use IP addresses to identify intermediate systems. Instead, OSI addresses are used. Although many people find OSI addresses confusing, this is more from lack of common use than because of the addressing scheme itself. In common use, the OSI address space has four parts: a *domain*, an *area*, a *system identifier*, and a *network service access point* (NSAP). There are actually more parts to the address space, but these are the four we tend to deal with for deployments of IS-IS within IP networks. Figure 4-2 illustrates the addressing scheme.

FIGURE 4-2 OSI Addressing Scheme



- The domain is almost always 49, which means this is a private domain, unless the company has been assigned a domain by a registry.

- The area is the part that confuses most people, because it is variable length. Most network designers keep the area to 1 octet, because there are rarely ever more than 255 areas in a network, and because this makes the address simpler to manage and read. Some networks do use a 6-octet area number, however, which just expands the address by one more section.

- The system ID is unique per device within an area, although it is simpler to keep it unique per device within the network, just for administrative purposes.

- The NSAP indicates a service located on the device. This should always be set to 0 when configuring IS-IS for IP routing. It is used to indicate a link-state protocol data unit (LSP) that represents a pseudonode.

The node address, made up of these four parts, is called the NET.
Any two routers in the same domain may form a neighbor relationship. Any two routers in the same area will share level 1 routing information (as long as the link between them is configured to transmit and receive level 1 routing information), and any two routers in different areas will share level 2 routing information (as long as the link between them is configured to transmit and receive level 2 routing information).

Path Selection

Each IS transmits information about neighbors and links it is connected to in an LSP. Figure 4-3 illustrates the information generated and received by an IS:

- **Router A would advertise a single LSP containing**

- A connection to Router B

- A connection to 10.1.1.0/24

- **Router B would advertise a single LSP containing**

- A connection to Router A

- A connection to Router D

- A connection to 10.1.1.0/24

- A connection to 10.1.2.0/24
- **Router C would advertise a single LSP containing**
- A connection to Router D
- A connection to 10.1.2.0/24
- **The DIS would generate a pseudonode LSP for Pseudonode D containing**
- A 0 cost connection to Router B
- A 0 cost connection to Router C
- A connection to 10.1.2.0/24

You should notice that the cost from the pseudonode to each connected node is always 0, whereas the cost to the pseudonode is reported as the link cost on the locally advertising router. This prevents the cost of a single link from being advertised twice, and permits links to have different costs in different directions (asymmetric costs).

Although each IS generates only one LSP, this single LSP is broken up into fragments. Each fragment is flooded independently of all the other fragments, so a single change in the network does not necessarily require the reflooding of the entire LSP, including all its fragments, but rather just some of the fragments within the LSP.

IS-IS uses Dijkstra's shortest path first (SPF) algorithm to determine the shortest path through the network. Each router transmits a copy of its local link-state database to its fully adjacent neighbors (this process is called *flooding* the database), so each router within a single flooding domain, or area, has the same set of LSAs in its local database.

IS-IS runs the SPF algorithm across this database to create an SPT. This SPT provides a loop-free path to each destination in the network, and thus provides the basis for the routing information IS-IS installs in the local routing table.

IS-IS uses a single opaque metric to describe the cost of a link. The cost each router advertises is the local cost configured on the link it is

advertising. It is possible for links to have different costs on each end (asymmetric costs), so that SPF returns different results from different points in the network. This is normally not a problem, as long as no links are 0 cost, other than the links to pseudonodes, as described earlier.

External routing information is described in the same way as internal routing information, but markers indicate the routing information is from an external source. Internally learned routes are preferred over information learned from external sources. IS-IS does not carry any information about the external routing domain, such as the original cost of the route from the other protocol.

An IS-IS router does not generate a new LSP when a link changes immediately, nor does it run SPF immediately on receiving new routing information. Rather, both of the operations have associated timers that help to dampen the pace of change in the network:

- The amount of time an IS-IS implementation will wait after noting a link or neighbor change until it transmits an LSP is variable, but is generally set to around 5 seconds. This timer can normally be adjusted through manual configuration, and can sometimes be set to back off exponentially to prevent a large number of LSPs from being generated quickly.

- The amount of time an IS-IS implementation will wait after receiving an LSP before running SPF is variable, but is generally set to around 5 seconds. This timer can normally be adjusted through manual configuration, and can sometimes be set to back off exponentially as LSPs are received more quickly.

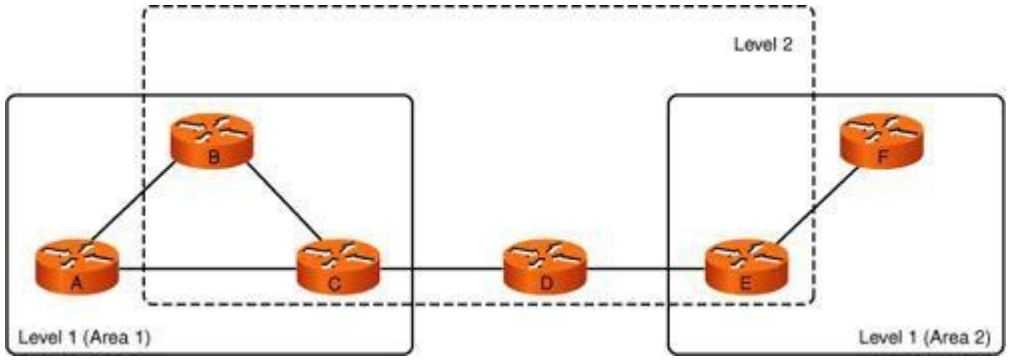
Network engineers would normally adjust the LSP generation and SPF timers to improve network convergence times. The tradeoff is as these timers are reduced, the network becomes less stable as the rate of changes increases.

Aggregation of Routing Information

IS-IS aggregates topology information by allowing the network engineer to break the network up into multiple flooding domains. There is a single level 2 flooding domain, and there may be one or more level 1 flooding domains. No relationship exists between the "network core" and the level 2 flooding domain, in a logical sense. The level 2 flooding domain is simply the flooding domain that interconnects the level 1 flooding domains.

Because IS-IS carries level 1 and level 2 routing information in different packets, forming independent neighbor adjacencies at each level, the level 2 flooding domain can overlay the level 1 flooding domains, as shown in Figure 4-4.

FIGURE 4-4 IS-IS Flooding Domains



In this network:

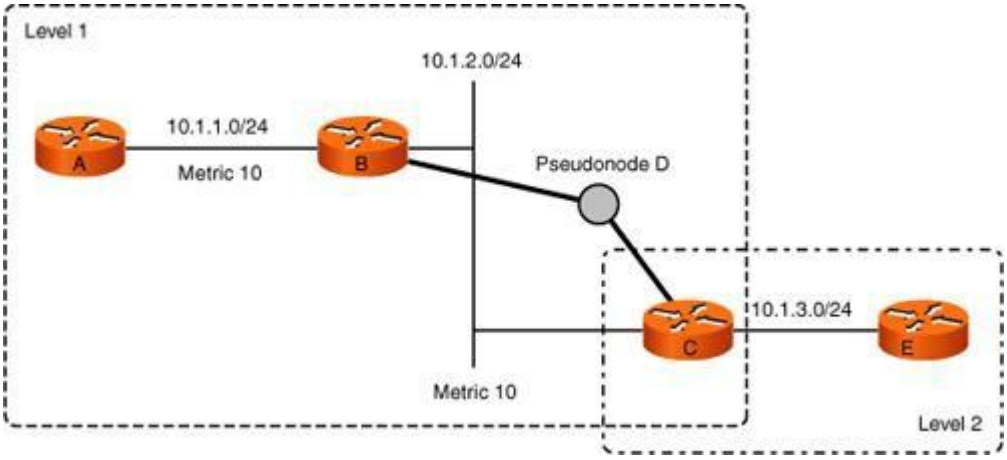
- Routers A and B are peering at level 1 within area 1 only.
- Routers A and C are peering at level 1 within area 1 only.
- Routers B and C are peering at level 1 within area 1 and level 2.
- Routers C and D are peering at level 2 only.
- Routers D and E are peering at level 2 only.
- Routers E and F are peering at level 1 within area 2 only.

Any router that has both level 1 and level 2 neighbors will summarize the information from its level 1 link-state database into the level 2 routing domain. To summarize, the IS finds the cost to each destination

within the level 1 routing domain, and inserts the destination with that cost into its level 2 LSP, as if the destination were directly connected with the cost associated being the cost to reach the destination through the level 1 routing domain.

Each IS with neighbors within a level 1 routing domain, and attached to a level 2 routing domain, sets the *attached* bit in its level 1 LSP. This indicates that level 1 only intermediate systems can forward traffic for unknown destinations to this router, which should have more information. Destinations contained in the level 2 routing database are not normally advertised into the level 1 routing database (although information can be leaked from level 2 to level 1 in some implementations), as shown in Figure 4-5.

FIGURE 4-5 IS-IS Flooding Domain Boundary Summarization



- **Routers A, B, and C have the following in their level 1 LSP databases:**

- An LSP from Router A containing

- A link to Router B

- A link to 10.1.1.0/24

- An LSP from Router B containing

- A link to Router A

- A link to Pseudonode D
- A link to 10.1.1.0/24
- A link to 10.1.2.0/24
- An LSP from Pseudonode D containing
- A 0 cost link to Router B
- A 0 cost link to Router C
- A link to 10.1.2.0/24
- An LSP from Router C containing
- A link to Pseudonode D
- A link to 10.1.2.0/24
- The attached bit
- **Routers C and E have the following in the level 2 LSP databases:**
- An LSP from Router C containing
- A link to Router E
- A link to 10.1.1.0/24
- A link to 10.1.2.0/24
- A link to 10.1.3.0/24
- An LSP from Router E containing
- A link to Router C

- A link to 10.1.3.0/24

Aggregation of reachability information and route filtering can be configured only on routers configured to run both level 1 and level 2 routing, toward the level 2 routing domain.

Chapter 5. BGP Theory and Operation

The Border Gateway Protocol Version 4 (BGPv4, or more simply, just BGP) is the only interdomain, or exterior gateway, routing protocol in wide use today. To understand BGP, we must start by understanding what BGP defines as a routing domain.

Note

A routing domain, within BGP, is a network, or a set of networks, which share a common set of administrative policies toward external networks. Although a routing domain may contain many networks with different internal routing policies, a routing domain always appears as a single abstract "cloud," with a single consistent routing policy, to outside entities.

The definition of a routing domain, within BGP, is intentionally fuzzy. A domain may be as small as a single router or as large a several thousand routers. The emphasis is always on the concept of a common policy toward outside networks, or rather, a common administrative and policy boundary.

A single corporate entity may have several departments, or business units, which act in generally independent ways, and yet may share some common set of network infrastructure, applications, or devices. Each of these departments or business units might appear as an independent routing domain, and hence, the best way to connect them might be using BGP. On the other hand, a single corporate entity might appear to be a single routing domain to all other networks, and thus might be internally connected using Interior Gateway Protocols (IGPs) (Enhanced Interior Gateway Protocol [EIGRP], Intermediate System-to-Intermediate System [IS-IS] Protocol, or Open Shortest Path First [OSPF] Protocol),

and may only connect to outside networks using BGP.
In BGP, a network contained within a single routing domain is called an *autonomous system*. Each autonomous system has an autonomous system number, which is unique within the internetwork. There are publicly assigned autonomous system numbers, which are globally unique, much like publicly assigned IP address spaces, and there are private autonomous system numbers, which are used only in networks that are not connected to the global Internet, or do not require unique identifiers.
BGP, therefore, is designed to connect routing domains under different administrative controls, or autonomous systems, and is generally concerned with policy rather than with fast convergence or other routing goals.
A device that is running BGP is called a *BGP speaker*. All devices that run BGP do not forward traffic; many devices run BGP for administrative purposes, such as route view servers, and are configured so that they are not in the path of packets being forwarded through the network. A pair of BGP speakers that have formed a neighbor relationship are said to have a *peering session*, or the speakers are *peered*.

Neighbor Relationships

BGP is unusual among routing protocols in that it is designed to run on top of an existing TCP connection. In many ways, BGP appears as an application running on top of an existing IP network, with an underlying IGP providing IP reachability between BGP speakers, and BGP providing IP reachability between networks. In a sense, BGP builds a virtual interdomain topology on top of a set of existing networks.
A pair of BGP speakers that have formed a peering session within a single autonomous system are said to have an iBGP session, and a pair of BGP speakers in two different autonomous systems that have formed a peering relationship are said to have an eBGP session.
BGP speakers peer with one another using TCP, which means BGP itself has no transport functionality. BGP does not discover neighbors; the

speakers with which it should peer are manually configured by a network operator. BGP has no reliable transmission system because BGP protocol relies on TCP for reliable transport. Important aspects to remember about BGP peering include the following:

- BGP speakers can build a peering session over a link that includes multiple IP routed hops. iBGP sessions can be formed across a network of multiple Layer 3 IP hops with no additional configuration. eBGP sessions generally require some form of special configuration to form a session across a multihop network.

- BGP speakers use the well-known TCP port 179 to open BGP sessions. When a BGP speaker opens a session to a peer, it opens the session from an ephemeral TCP port, and to TCP port 179. If both BGP speakers open a session at the same time, there is a mechanism in BGP to tear down one of the two sessions.

- The BGP speaker that is using a TCP ephemeral port as its source port is said to be the active speaker, and the speaker using TCP port 179 as its source port is said to be the passive speaker. Some implementations of BGP allow the speaker to be configured so that it never opens a session, but rather only waits for peering speakers to open sessions. This can be an important capability in some large-scale deployments of BGP.

- Because BGP runs over TCP, and can form peering relationships over multihop IP networks, BGP can be configured to run through devices such as firewalls and gateways.

BGP speakers send routing information to another speaker using an *update*. The way this update is organized, internally, plays a large role in the scaling and convergence properties of BGP. Figure 5-1 illustrates the basic structure of a BGP update.

FIGURE 5-1 The BGP Update Format



As you can see from this illustration, BGP carries two pieces of information within a routing update:

- Attributes, which are information about routes and destinations. This is metadata that describes the paths to a given destination, contains policies relating to the reachability of a given set of destinations, or contains information that groups a set of destinations together.

- Network Layer Reachability Information (NLRI), which contains destinations within the network this BGP speaker knows how to reach.

A route, within BGP, is a destination that is reachable within the network, combined with the attributes describing that destination, or policies about the speaker's ability to reach that destination. If you examine the structure of the BGP update carefully, you will notice that each NLRI does not carry its own set of attributes. Instead, the update contains a set of attributes that apply to all the destinations listed in the NLRI section of the update.

This implies that as the number of attributes grows, the number of destinations any given set of attributes applies to will decrease, and hence, the larger number of packets BGP must send to transmit a full table of routes between two speakers, and the slower BGP will converge.

It is also important to consider that BGP uses a unicast transport mechanism to send update packets. This means a BGP speaker with 1000 peers must build and format 1000 unicast packets, even if the update being sent to all the peers is identical. Most implementations use various complex techniques to generate a single update that can be transmitted to as many peers as possible, to reduce the overhead of packet transmission on the BGP speaker.

These mechanisms, however, rely on the peers to which the single transmitted update is being sent sharing a common set of policies, so the same set of attribute/NLRI sets is transmitted to all the peers. Complex outbound policies defeat these algorithms, causing BGP to converge more slowly.

BGP has simple rules for forwarding routes between peers:

- If a route is learned from an iBGP peer, it is transmitted only to eBGP peers.

- If a route is learned from an eBGP peer, it is transmitted to all peers.

These rules generally mean that all the iBGP speakers within an autonomous system must be peered to one another. Some exceptions to these rules apply, as discussed in "Route Reflectors" section.

Most BGP implementations do not require a route to be installed in the local routing table by the BGP speaker for the BGP speaker to advertise the route. For instance, if a route to 10.1.1.0/24 has been learned through both OSPF and BGP, and OSPF installs the route in the local routing table, the BGP speaker will still advertise the route to 10.1.1.0/24 to both iBGP and eBGP peers. This behavior can normally be controlled through manual configuration.

Path Selection

The BGP path-selection algorithm is complex, and is becoming more complex over time as new capabilities are added to the protocol. In general, however, the entire BGP path-selection process can be summed up in a small number of steps:

- Make certain the path being considered is not a loop.

- Make certain there is an IP path to the next hop toward the destination.

- Use locally preferred exit points out of the network first.

- Use the entry point the peering autonomous system would prefer next.

- Use the closest exit point if all else is equal.

BGP uses the route attributes to make a decision at each of these points.

Each attribute is described here, in the order in which it is used in the BGP decision process:

- To determine whether a path is loop free, the BGP speaker examines the *autonomous system path*. The autonomous system path is a listing of the autonomous systems through which the routing information has traversed to reach the current BGP speaker. When advertising a route to an eBGP peer, a BGP speaker adds its local autonomous system to the autonomous system path. If the local autonomous system appears in the autonomous system path, the route is discarded as a loop. The autonomous system path is a transitive attribute, which means it is carried between eBGP speakers.

- To determine whether there is an IP path to the next hop, the BGP speaker examines the *next hop*. The next-hop attribute contains the IP address traffic transmitted to this destination should be sent to. If there is no IP path to the next hop, the route is discarded. The next-hop attribute is generally set to the local address of the BGP speaker when a route is being transmitted to an eBGP peer, and left intact when a route is being transmitted to an iBGP peer. Most implementations allow this behavior to be overridden, however.

- To determine whether the route is locally preferred, the BGP speaker examines the *weight*. The weight is not carried within the BGP update, and is generally not considered an attribute in the true sense. The weight is a local configuration on the BGP speaker that allows that specific speaker to be configured so that it prefers routes in a particular order.

- To determine the most preferred exit point out of the local network, the BGP speaker examines the *local preference*. The local preference is carried in the BGP update packet, but only within the autonomous system. Because the local preference is reset when a BGP update is being transmitted between eBGP peers, it is considered a nontransitive attribute.

- To determine the most preferred entry point into the peering autonomous system, the BGP speaker examines the *multiple exit discriminator* (MED). The MED is normally set manually or taken from the IGP metric when a route is transmitted between eBGP peers. Because the MED is transmitted between eBGP peers, it is considered a transitive attribute.

- To determine the closest exit point, the BGP speaker examines the IGP metric. The IGP metric is not an attribute of the BGP route itself, but is rather taken from the cost to reach the next hop as shown in the local routing table.

- If all these metrics are equal, BGP uses a series of possible tiebreakers to determine which route to use. The BGP specification states that the route learned from the peer with the lowest router ID should win, but some implementations choose the winning route by examining the age of the route and choosing the older route.

BGP has the capability to set the next-hop attribute to a third party, which means the next hop may be set to a router the local router can reach, or that shares a broadcast interface with the transmitting router. This is normally not used, but it is available within the protocol itself and most implementations.

Aggregation of Routing Information

BGP allows for the aggregation of routing information at any point in the network, whether across iBGP or eBGP connections. Normally, however, to maintain a consistent routing policy throughout an autonomous system, it is good design practice to aggregate only at eBGP connections.

Important points to consider when using BGP aggregation include the following:

- Most BGP configurations do not automatically block the advertisement of longer-prefix routes when aggregating routing information. Normally, BGP will advertise both the shorter aggregate prefix and the longer prefixes within the shorter prefix unless the speaker is manually configured to block the longer component prefixes.

- BGP aggregates the autonomous system paths of the component prefixes within an aggregate by creating an autonomous system set object (as-set). An as-set is a set of autonomous systems through which this routing information may have passed, but no ordering is implied, nor is it implied that every destination reachable in the aggregate is reachable through the set of autonomous systems in the as-set.

BGP aggregation is actually quite rare; it is not common to see BGP aggregation configured on a BGP speaker. This does not mean routes are not normally aggregated in a large internetwork, like the Internet, but rather that other mechanisms are used for aggregating routing information.

The most common mechanism used to aggregate routing information is a combination of manually configured static routes combined with a route filter. To aggregate 10.1.0.0/24, 10.1.1.0/24 10.1.2.0/24, and 10.1.3.0/24, for instance, the most common configuration would be this:

- Create a static route for 10.1.0.0/22 on the local BGP speaker, and distribute or redistribute this route into BGP.

- Create a filter permitting only 10.1.0.0/22 within the 10.1.0.0/22 address space, and apply this filter toward all eBGP peers.

Why is this normally used rather than BGP aggregation (as described in the BGP specifications)? There are a number of reasons:

- It prevents the constant changing of the as-set when various components of the aggregate change state. If the point of aggregation is to hide the state changes, aggregating in a way that causes the shorter-prefix route to change each time the longer-prefix components change is not extremely helpful.

- It allows the network operator to deterministically preset network behavior in specific network conditions.

- It allows the network operator to treat the aggregate as any other route, using standard tools to influence inbound route preference with peering autonomous systems.

Policies

BGP policies generally relate to the following:

- Choosing the best exit point out of the local autonomous system

- Influencing a peering autonomous system to choose the best path into the local autonomous system

- Controlling the advertisement of routing information into peering autonomous systems

- Controlling the advertisement of routing information by peering autonomous systems

Some of these policies are implemented using a BGP attribute we have not yet discussed here, BGP communities. The following subsection discusses BGP communities, and then the other subsections discuss some common techniques for implementing these policies.

BGP Communities

BGP communities are designed to group a set of routes into a common policy, but are more often explained and understood as simple tags on routes. BGP can carry two classes of communities: standard and extended. A standard community is 4 octets, and an extended community is 8 octets.

Extended communities are encoded in two parts, with the first part being

either 1 or 2 octets, depending on the type of extended community. Extended communities are written using the notation XX:XXXX, where the part before the colon indicates the type of community, and the part after the colon indicates the actual community value.

Controlling the Exit Point

Controlling the exit point is generally the simpler problem to solve in the routing policy realm because the packet is currently under the control of the local network (the traffic being directed is under local control). The points at which the outbound exit point can be influenced in the BGP decision process are as follows:

- **The weight**: This control point is on the router for which weight is locally configured only, but it is still useful in many situations. For instance, if the local autonomous system is learning the same route from two different autonomous systems through two different BGP speakers, the weight can be used to prefer the locally learned route over all other routes, so a single BGP speaker always prefers a local exit point.

- **The local preference**: This control point allows the network engineer to direct traffic from within an entire autonomous system toward a single exit point out of the autonomous system. For instance, the local preference is often used by service providers to prefer routes to their customers through the connections to those customers over routes to their customers learned from peers.

- **The IGP metric**: By tuning the metrics of the underlying OSPF, IS-IS, or EIGRP implementation, a network engineer can steer traffic in a network to specific exit points, by adjusting the cost of the path to the next hop.

Some implementations of BGP also support the ability to use a community string to prefer a specific exit point, through the *cost community*.

Controlling the Entry Point

A network engineer can influence the inbound path of traffic destined to a specific destination in four ways. The first (most obvious, and often the least effective) is to set the MED when advertising routes toward eBGP peers.
The MED is generally not effective because

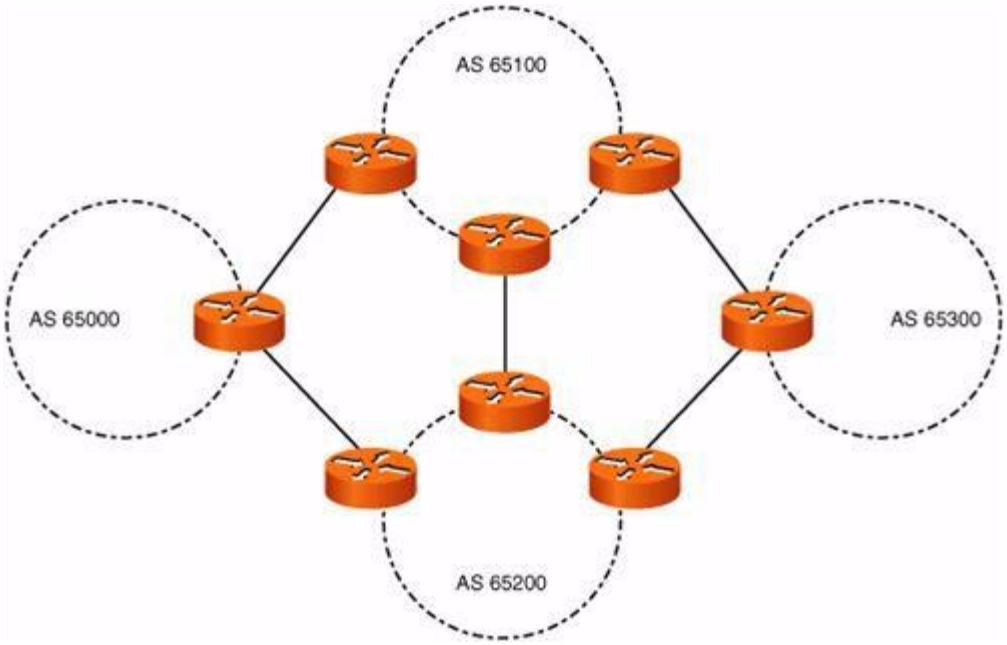
- The MED is compared only if the autonomous system path on the two routes being compared is identical. This means the MED is generally useful only if the two peers the routes are being advertised to are the same autonomous system.

- The MED is lower in the BGP decision process than the local preference. The peer to which the two routes are being advertised must be setting the local preference the same on both routes for the MED to have any impact on the BGP decision process.

- The MED is sometimes stripped by the peer. It is simple to configure BGP to ignore the MED, or strip it from the attribute set, or reset it to some standard value, when receiving routes from an eBGP peer. Many service providers reset or strip the MED as part of their standard operations.

The second most common technique used to modify the inbound path is modifying the autonomous system path, by prepending additional autonomous systems onto the autonomous system path. Although this is sometimes useful, it does not generally have the impact you might imagine when configuring autonomous system path prepending. Figure 5-2 illustrates why this is.

FIGURE 5-2 Autonomous System Path Prepend



Assume the network administrator for AS65000 would like more traffic to come in through the link to AS65100 than through the link to AS65200. To accomplish this, the network administrator prepends AS65000 onto the autonomous system path of the routes being transmitted to AS65100, and leaves the autonomous system path of the routes being transmitted to AS65200 alone.

The assumption is that AS65100 will end up preferring the path through AS65200, rather than the AS65100 to AS65000 link. Normally, however, AS65100 would use local preference to prefer the path directly to AS65000, rather than any path learned from a peer. Hence, autonomous system path prepend does not do anything to impact the flow of traffic from either of the two directly connected autonomous systems, AS65100 or AS65200.

The one place where autonomous system path prepend may have an impact is at AS65300, which does not treat AS65000 as a customer, and so, most likely, treats all routes learned to destinations within AS65000 equally, at least in terms of local preference. AS65300, however, might have other local policies that will defeat the autonomous system path prepending.

Overall, autonomous system path prepend might or might not impact actual traffic flows into an autonomous system.

RFC 1998 describes another option that may be used to control the point traffic enters into an autonomous system. The essential idea is that a

BGP speaker can attach a community that the service provider uses to set the local preference of received routes. Returning to Figure 5-2, this means AS6500 would transmit its routes toward AS65100 with a community instructing AS65100 to set its local preference to some value that causes AS65100 to prefer the routes through AS65200. Although some service providers support this use of communities, they generally will not allow their customers to go against the service provider's best economic interest, which generally involves using their links to their directly connected customers as heavily as possible.
The final, and generally most effective, mechanism a network engineer can use to influence the entry point of traffic into an autonomous system is to break aggregated routes up into longer-prefix components and advertise these longer-prefix routes out a subset of the available peering points.

Controlling Route Advertisement

BGP also provides the ability to control, through communities, where a route advertised to another autonomous system is advertised. Several well-known communities are specified, including the following:

- NO_EXPORT, which means the receiving peer should not advertise this route outside the receiving autonomous system
- NO_ADVERTISE, which means the receiving peer should not advertise this route to any other peer, within or outside the receiving autonomous system

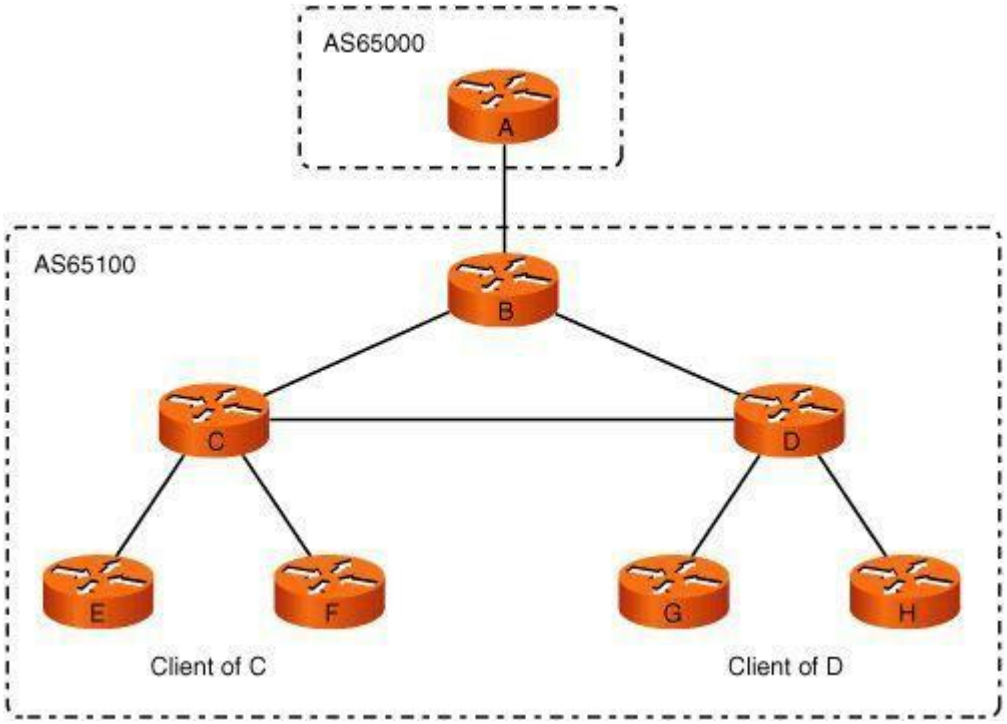
A number of service providers also have a number of other communities a customer can set to impact the advertisement of a given route.

Route Reflectors

BGP requires that all iBGP speakers be peered to all other iBGP

speakers within the same autonomous system. This impacts BGP scaling within an autonomous system, making the configuration and management of iBGP very difficult. Route reflectors provide a scaling mechanism for large-scale iBGP deployments. Route reflection adds an additional attribute to BGP routes, which effectively builds a path of the route reflectors the route has passed through within the autonomous system, which serves the same purpose as the autonomous system path serves between autonomous systems. Figure 5-3 illustrates the operation of route reflectors.

FIGURE 5-3 Route Reflector Operation



When Router B receives a route from Router A, it uses normal iBGP processing rules to send the route to Routers C and D. Let's follow the update as it passes through Router C. Router C, on receiving this route, adds two new attributes to the advertisement:

- An *originator ID*, which contains Router B's router ID, as the first BGP speaker within this autonomous system to receive the route.

- A *cluster list*, which contains a list of the route reflectors through

which the route has passed. In this case, Router C adds itself as the only entry on the cluster list.

Router C then advertises this route to Routers D, E, and F (Router C is *reflecting* the route rather than advertising it by adding these attributes). When Router D receives this reflected route, it adds itself to the cluster list, too. Router D determines it should not advertise the route to Router B, because Router B is in the originator ID field. Router D will, however, reflect the route to Routers G and H.

Key factors to consider when deploying route reflectors include the following:

- Route reflection, by removing information from the routing system, can often result in a suboptimal route being chosen, or rather a suboptimal exit point from the autonomous system.

- A route reflector client should never be peered to a route reflector through another route reflector, to prevent loops within the autonomous system.

- Route reflectors can be deployed in a hierarchical manner, so that a route reflector itself is a client of a higher-level route reflector. The number of levels within this hierarchy should be kept to a manageable level.

- Route reflector cluster IDs need to be chosen with care. A route reflector will reject routes with the local cluster ID in the cluster list. This is important from a resiliency perspective; multiple clusters with different cluster IDs will increase the number of routes in the routing table of any given BGP speaker within the autonomous system, but it also provides resiliency.

Chapter 6. General Routed Design Theory

Routed network design is at the heart of the CCDE. Understanding the concepts and principles of Layer 3 design, focused on routing, is critical to achieving the CCDE certification. This chapter discusses the general principles of routed network design. Included are sections on route aggregation, fate sharing, redundancy and resiliency, convergence tuning, configuration complexity, and multicast design principles. Then Chapter 7, "Topology Design Theory," discusses specific topologies and how they interact with the routing protocols.

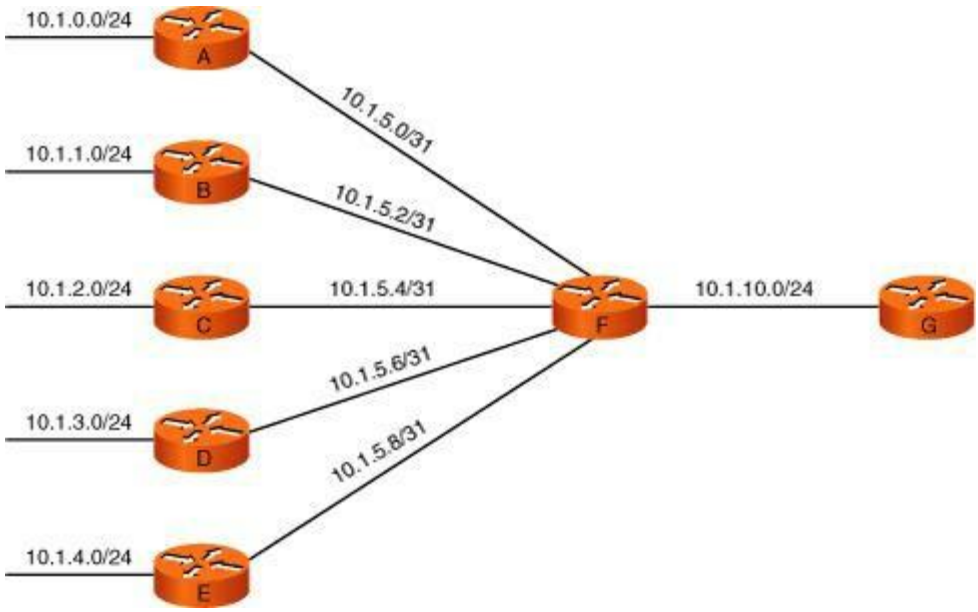
Route Aggregation

Route aggregation serves two specific purposes in a network:

- Breaking the network into multiple failure domains
- Reducing the amount of information the routing protocol must deal with when converging

Figure 6-1 will be used to illustrate both of these principles.

FIGURE 6-1 Aggregation Principles



Based on the IP addressing within this network, it is possible to aggregate at Router F toward Router G to 10.1.0.0/22, which would provide reachability to 10.1.0.0 through 10.1.7.255. Aggregation at Router F breaks up the failure domain by blocking notifications or updates about individual link failures between Routers F and A, B, C, D, or E from being transmitted to Router G. For instance, if the link between Routers A and F fails, the aggregate Router F is sending to Router G does not change.

Aggregation at Router F can also improve network convergence by decreasing the amount of information routers in the network must process. If a new router is attached to Router G, it needs to send just 2 routes, rather than 11. There is a large multiplier effect as a network grows.

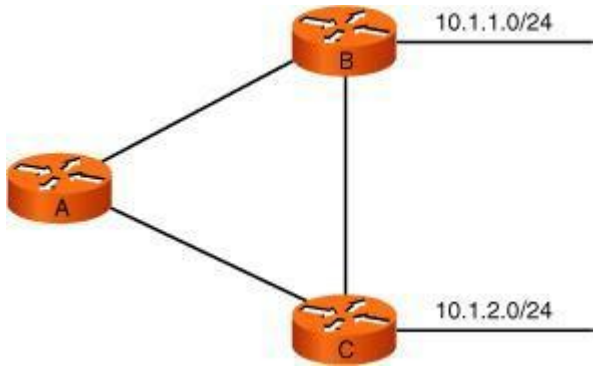
Most routers build a *discard route* when advertising an aggregate. The discard route causes the traffic falling within the aggregate route, but not within one of the more specific routes in the aggregating routers routing table, to be discarded. For instance, if Router G transmits a packet toward 10.1.6.1, the only route in Router F's routing table that will match this destination is the discard route built off the aggregate Router F is advertising toward Router G.

Network engineers should always be aware that the longest prefix within the local routing table that contains the destination address will be used for forwarding. For instance, if a packet is received with a destination address of 10.1.1.1, and there are two routes in the local routing table

(10.1.1.0/24 and 10.1.1.0/25), which both contain, or could provide a valid route, to the destination, the route with the longer prefix length will always be preferred over the route with the shorter prefix length. This might appear to be a simple rule, but this simple rule will often cause unexpected misrouting of traffic.

The longest prefix rule can also be used as an advantage. For instance, in the network in Figure 6-2, you could use the longest-match rule to optimally route traffic to the correct destination, while allowing for a backup path that is always in the local routing table, for faster convergence.

FIGURE 6-2 Longer Prefixes and Optimal Routing



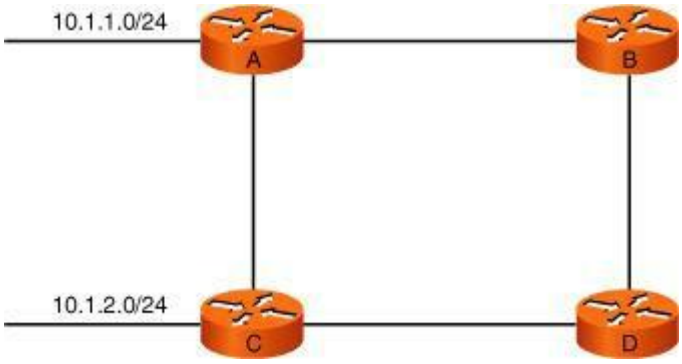
If the following advertisements are configured in the network

- Router B advertises 10.1.0.0/16 and 10.1.1.0/24 toward Router A.
- Router C advertises 10.1.0.0/16 and 10.1.2.0/24 toward Router A.

For packets destined to 10.1.1.1, Router A will choose the path through Router B, whereas for packets destined to 10.1.2.1, Router A will choose the path through Router C. If the Router A to Router B link fails, however, Router A will choose the path through Router C to reach 10.1.1.1, because that is the only path available to this destination.

In many situations, the discard route built through the configuration of an aggregate route will create a black hole in the event of some specific link failures, as shown in Figure 6-3.

FIGURE 6-3 Aggregate Black Holes



If the following routes are being advertised in this network

- Router A is advertising 10.1.0.0/16 to Routers B and C.

- Router C is advertising 10.1.0.0/16 to routers A and D.

If the link from Router A to Router B fails, Router D will forward traffic destined to 10.1.1.1 to Router C. Because Router C has an aggregate route to 10.1.0.0/16, and no more specific route to the destination, it will forward the traffic to the discard route created when the aggregate was configured. Although a physical path to the destination is available, the traffic is discarded because of the way the aggregation is configured. In general, you should always have a link between any set of routers that are configured with the same aggregates on which aggregation is not configured.

Network engineers should also be careful of the following when configuring or using aggregated routing information:

- Aggregate routes can bring traffic further into the network than you might want, which could result in security issues. For instance, packets resulting from an attack that are directed to a destination that does not exist in your network could pass deep into your network before they are discarded at the router with a discard route because of an aggregate.

- Most routing protocols take the metric for an aggregate route from the component routes, or the routes that are blocked by the aggregate route. Routing protocols take the metric from the component route with the lowest metric or the highest metric, and

advertise the aggregate route with the same metric. If the route from which the aggregate's metric is taken fails, or is withdrawn, the aggregate's metric will change, too, communicating the change to the routers beyond the aggregate route. This defeats the purpose of the aggregate route. Network engineers should consider this when designing route aggregation, using techniques to keep the aggregate's metric from changing where possible.

Fate Sharing

Network engineers generally consider fate sharing only when there are multiple logical signals on a single physical wire, but the concept of fate sharing, and its application in network design, is much broader than this specific case. Any time a network is virtualized, fate sharing will be the result. For instance:

- Multiple wavelengths over a single fiber, using dense wavelength-division multiplexing (DWDM)
- Multiple circuits multiplexed through add/drop multiplexers on a SONET link
- Multiple circuits running over a single Frame Relay circuit
- Multiple VLANs running over a single physical Ethernet cable
- Multiple VPNs running over a single Layer 3 infrastructure

Each time a network engineer sees any sort of virtualization being used, they should look for fate-sharing points and consider designs that will counter fate-sharing failures in the network.

Fast Convergence

Many modern networks require short convergence times to support various applications, such as voice and video. It is best to break convergence time in three pieces and consider each piece separately:

- The time required to detect the network failure that requires convergence
- The time required to find an alternate path through the network
- The time required to install the alternate path in a device's local forwarding table and begin forwarding traffic along the alternate path

A network engineer can do little about the time required to install an alternate path in any given device's local forwarding table, so this aspect of fast convergence is not covered in this section.

Detecting Network Failures

The detection of network failures consumes most of the convergence time budget in typical designs. There are several considerations in this area, including the following:

- Event drive detection of link or neighbor failures is almost always going to be faster than polled detection of these failures. For instance, detecting the loss of carrier on a point-to-point Ethernet link is always faster than detecting the loss of three "hello" or "status" packets no matter how fast those hello packets are transmitted, received, and processed.
- Network engineers need to consider the properties of the underlying logical media, such as what the impact of specific failure modes might have on Layer 3 connectivity. For instance, if it is possible for a link to fail in a way that allows multicast traffic, but not unicast, or to fail in a way that allows traffic to pass in only one direction along the link, these are failure modes

that need to be taken into account in the choice of protocols running over the link, and the network design at large.

Finding an Alternate Path

When a network failure has been detected, the routing protocol must find an alternate path to the destination. A network engineer needs to keep in mind several factors in this area:

- The fastest possible convergence is always achieved when there are alternate available paths already installed in the forwarding table of the device that detects or is adjacent to the network failure. This is not always possible, however, and network engineers should be careful when adding redundancy to achieve this effect (see the "Redundancy and Resiliency" section).

- There are many techniques developed for routing protocols specifically designed to improve network convergence times. Many of these techniques can produce extremely fast convergence, often at the cost of added complexity. Network engineers should strive to reach a balance between improved convergence and network complexity. Use the minimal set of capabilities possible to achieve the design goals.

- Techniques that improve network convergence may also decrease network stability. Network engineers need to balance between network convergence speed and network stability; negative feedback loops in the routing protocol are the most common cause of large-scale network failures.

- The primary causes of slow network convergence are interactions with underlying transport mechanisms (where slow failure detection slows down network convergence) and the design of the network (which is either too complex for the requirements at hand, does not match with the routing protocol in use very well, or does not break up failure domains and hide information well

enough). Failure detection and fundamental design are the two areas a network designer should examine first when considering how to improve network convergence.

Redundancy and Resiliency

In many cases, network operators equate redundancy and resiliency. To add resiliency, a network engineer will often just add another parallel link, or another parallel piece of equipment. However, adding redundancy does not always add resiliency.

One example of this is rather straightforward: Each time a link or piece of equipment is added to a network to create a parallel path, it reduces the total aggregate downtime due to failures. At the same time, each additional parallel path adds some finite amount of complexity that must be dealt with while troubleshooting any given problem and increases the network's convergence time by some amount. The total combination of additional troubleshooting complexity and longer network convergence can easily overcome the additional resiliency added to the network through the parallel path.

The process to discover exactly when adding a specific link or piece of equipment will result in more total downtime per year than less is complex, but the simple rule of thumb is the breakover point is generally reached someplace between the third and fourth redundant path.

It is also possible to add complexity and redundancy without adding resiliency in specific topologies. For instance, adding a second layer on top of an existing full-mesh network will add little resiliency, but it will add massive new complexity and redundancy. Adding a third ring to a dual-ring topology will rarely do anything to increase resiliency.

Once two or three parallel paths are available throughout the network, there is little gain from adding more parallel paths. It is generally better to consider alternative measures to increase resiliency, such as tuning routing protocol parameters, improving the speed of the data link to IP and routing protocol interaction, and other mechanisms.

Configuration Complexity

Network devices and protocol tend to provide multiple ways to solve a single problem or to deploy a particular feature. With increasing complexity in the application space, increasing virtualization, and increasing equipment capabilities, device configurations have become difficult to read and understand. Network engineers need to counter this trend by paying careful attention to network device configurations and

- Simplify configurations where possible.
- Configure with intent, which means choosing the most explicit configuration possible to accomplish a given task. For instance, instead of using Border Gateway Protocol (BGP) synchronization to prevent the advertisement of locally generated routes, use explicit filters. Do not count on secondary effects to provide primary effects, where possible.
- Apply policy and configuration as close to the point of use as possible.

Network engineers should also work to minimize the number of features and application-specific configurations used in the network to reduce complexity, make management simpler, and make troubleshooting easier.
Configuration management systems can be very useful, but network engineers should not assume that using a configuration management system is the final solution for complex configurations. Configuration management tools can configure only what network engineers tell them to configure. Further, when troubleshooting a network, you will often be working with raw configurations, rather than a configuration tool.
Redistribution between routing protocols requires special care in configuration complexity. Redistribution is a commonly used feature of routing protocols, and it is one of the features that can easily cause major problems, such as permanent routing loops. When redistributing between routing protocols

- Always use filters to prevent route leakage between multiple

redistribution points.

- Always use the simplest and most automatic filtering mechanism possible when filtering redistribution.

- Consider whether redistribution points should be policy division points (and route redistribution points). If the redistribution point connects two networks in different administrative domains, redistribution between two Internet service providers (ISPs) is generally not a good design decision. It is better to either redistribute static routing information or use a protocol designed to connect different administrative domains, such as BGP.

- Redistribute static or fixed routing information where possible, rather than live routing information.

- Make certain route metrics are accounted for in the redistribution design. Most routing protocol implementations require the redistributed metrics to be manually configured, because the metrics used by different routing protocols cannot normally be made equivalent.

Multicast Design Concepts

Multicast design is similar to unicast routing design in many respects, and different in other respects. Primary things to consider when deploying multicast include the following:

- When considering the placement of rendezvous points (RPs) for Protocol Independent Multicast (PIM), consider the traffic flow through the network, and consider the availability of the RP service under a wide variety of conditions.

- Use multiple RPs with anycast to provide resiliency.

- Multicast Source Discovery Protocol (MSDP) should always be used to connect multiple RPs that are used to manage the same set of multicast groups.

- Multicast traffic flow should be carefully considered and managed. For instance, multicast packets must be replicated someplace. Generally, it is better if the replication occurs through broadcast mediums closer to the destination, rather than closer to the source.

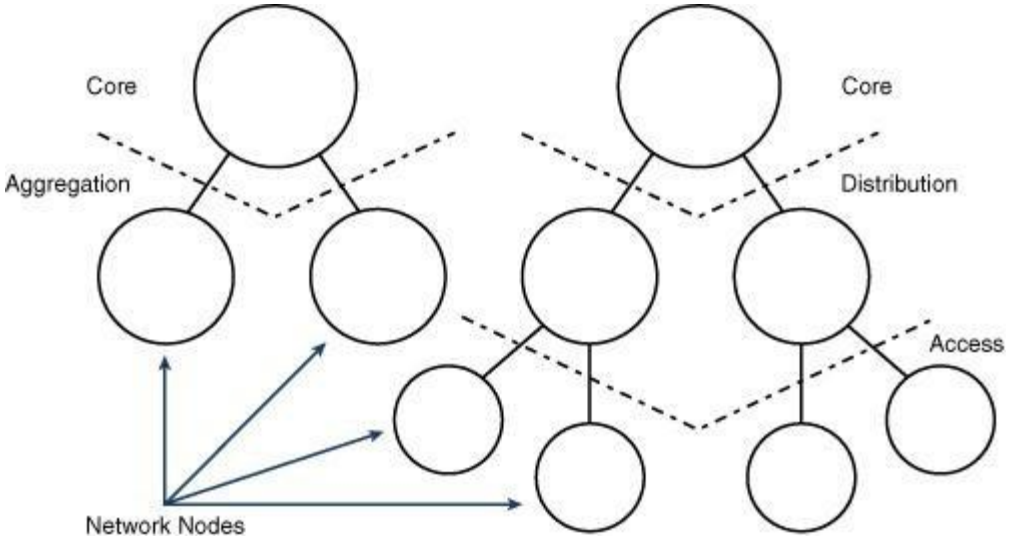
Chapter 7. Topology Design Theory

One of the most important considerations for routed network design is the topology of the network, or how the network is connected together physically and logically. The topology of the network determines the ability of the network designer to break a network into failure domains and aggregate routing information. Each section in this chapter briefly considers one aspect of topology in network design.

Essential Hierarchical Designs

Hierarchical network design is a widely explored area. This section discusses some of the fundamental areas of interaction between hierarchical network design and routing protocols. It is important to take a higher-level view of the hierarchical layers when examining network design. Although we traditionally speak of layers, a layer is really a set of "nodes" within the network within certain layers. Figure 7-1 illustrates this concept.

FIGURE 7-1 Hierarchical Network Components



There are two widely used hierarchical designs:

- **A two-layer design**
- A set of nodes within the aggregation layer, which provides for a place to connect service endpoints
- A core, which connects the aggregation nodes together
- **A three-layer design**
- A set of nodes within the access layer, which provides for a place to connect service endpoints
- A set of nodes within the distribution layer, which provides for a place to aggregate traffic, aggregate policy, and connect the access nodes to the network core
- A core, which connects the distribution nodes together

Network nodes are not connected within their layer, but rather only forward traffic through a higher level to reach another node within the same layer.

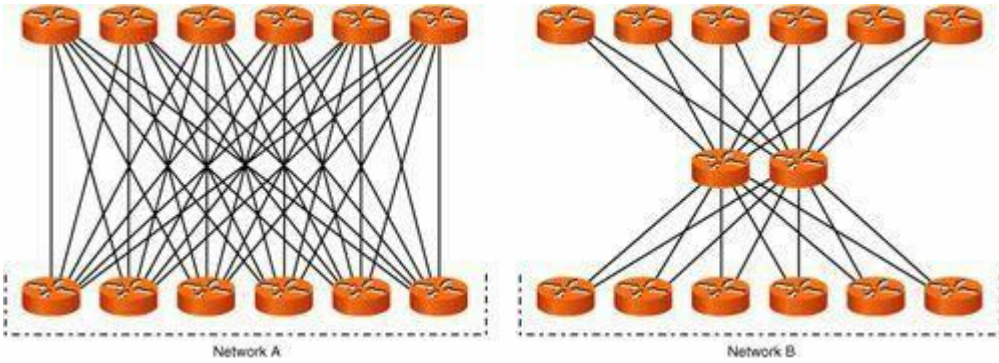
What advantages does designing a network in layers provide?

Fault Isolation

Separating a network into multiple failure domains typically involves the aggregation, or hiding, of routing information, as described in Chapter 6, "General Routed Theory Design." How does route aggregation break the network into multiple failure domains, providing for fault isolation? By reducing the number of network devices that are impacted by a change in the state of a link or device.

Designing in layers is important because it provides a natural boundary along which route aggregation, or other forms of information hiding, can take place, as shown in Figure 7-2.

FIGURE 7-2 Designing for Aggregation



In Network A in Figure 7-2, there is no logical or obvious place to aggregate routing information, or to apply policy. To reduce the amount of routing information on the top row of routers in Network A, a network engineer would need to configure aggregation on every router on the bottom row, on every interface toward the remote sites. This would require a great deal of work and make the network very difficult to manage.

In Network B in Figure 7-2, two new routers have been inserted, providing an obvious and natural pair of devices on which information can be aggregated or hidden, as needed. These two new routers form a "chokepoint" within the network.

Designing a network in layers creates the chokepoints required to effectively hide or aggregate information. The routers in the middle of Network B in Figure 7-2 provide a point at which faults can be isolated between the upper and lower rows of routers, so device and state changes on the bottom of the network do not impact the devices at the top

of the network.
These chokepoints also help to make troubleshooting easier in the network, by providing a place to start. If a problem were to occur in Network A in Figure 7-2, there would be no obvious place to start troubleshooting. Starting on any router is just as good as starting on any other, and it is possible that you would need to check every router in the network to find the problem, or even to find the symptoms of the problem.
In Network B in Figure 7-2, the routers in the center offer an obvious place to begin looking for network problems. Here is where you could check to see whether packets or flows are passing through the chokepoint between the two sections of the network, or check to see whether routes are being transmitted and received correctly. Checking the two center routers splits the problem between the top and bottom rows of routers.

Policy Separation

In the same way the chokepoints between hierarchical network components provide aggregation and information hiding points, they also provide points where policy can be implemented, and where policies within one part of the network can be divided from policies implemented in other parts of the network.
Some common policies implemented at the chokepoints between hierarchical layers include route filters, traffic engineering through routing metrics, packet re-marking, traffic policing, traffic shaping, and other mechanisms.

Choosing a Hierarchical Design

How many layers should you design around in a given network design? Where should you place layer boundaries? The general rule of thumb is to divide complexity from complexity where possible. Beyond this, use the minimal amount of hierarchy that provides the separation points required to effectively deploy and manage the network (and considers or

takes into account network growth).
Networks with a more meshed design, contain a smaller geographic area, consist of links of somewhat homogenous speeds, and over which applications requiring a high degree of any to any connectivity tend to fit two-layer hierarchies better. Networks covering a larger geographic area have a wider variety of link speeds, and are internally more complex, and tend to fit three-layer hierarchies better.
The number of hierarchical levels within a network can be deceptive. For instance, many service provider networks are modeled with two-layer hierarchies. In reality, there is a third layer within most service provider networks, consisting of the customer networks connecting to the service provider network. In some cases, networks may have subnetworks that each contain three layers, and are connected by a "core of cores," which are above, or outside, the three typical hierarchical layers.
In the end, the best design decision is to choose the optimal number of layers to separate complexity from complexity, and to provide the chokepoints needed to properly build isolated fault domains.

Routing Protocol Interaction with Specific Topologies

Network engineers also need to consider the interaction of the routing protocol to specific topologies within the network. Each of these topologies may represent only a small part of the network, but their interaction with routing can cause them to have a large impact on the overall network performance.

Full-Mesh Topologies

In a full-mesh topology, every device has a direct, one-hop (at Layer 3) connection to every other device within the topology. The number of connections is easily computed using the following formula:

Connections = (*devices* - 1) * (*devices*)

Hence, to connect 100 devices, you would need 990 links. Full-mesh topologies are complex in two different ways:

- The number of neighbor relationships each device must maintain
- The amount of information flooded through the network

For that same full mesh of 100 devices, each device must:

- Maintain 99 neighbor relationships, including a large amount of state
- Replicate each routing update and multicast packet into 99 packets, 1 for each neighbor in the full mesh

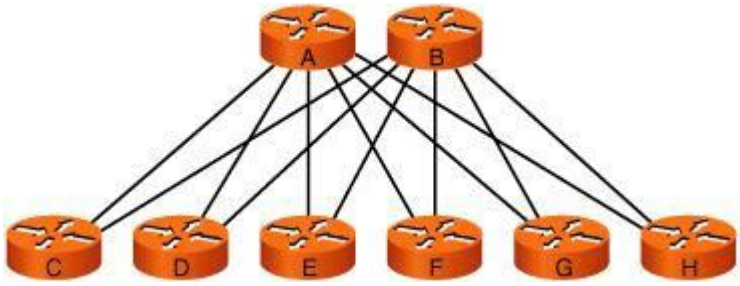
In many situations, full-mesh topologies, themselves, are not a scaling problem (although in some situations the scale of the full-mesh topology itself presents scaling problems). Most of the problems related to full-mesh topologies relate to the amount of duplicated information transmitted across the full-mesh topology and the impact failures within a full-mesh topology can have on other parts of the network. A network engineer can mitigate these issues with full-mesh topologies in several ways, including the following:

- Separate full-mesh topologies into their own failure domain through aggregation or information hiding. This is just a more specific instance of the general rule to separate complexity from complexity where possible.
- For link-state protocols, use flooding-reduction techniques where needed to meet the scaling requirements of the full-mesh topology. Flood-reduction techniques, such as *mesh groups* in Intermediate System-to-Intermediate System (IS-IS) Protocol, allow the network engineer to choose two or more designated flooders, blocking flooding by the other devices on the mesh topology.

Hub-and-Spoke Topologies

Hub-and-spoke topologies are more common than full-mesh topologies in real-world network designs. In a hub-and-spoke network, all the nodes are connected to a small number of hub nodes, generally one or two, as shown in Figure 7-3.

FIGURE 7-3 A Hub-and-Spoke Network



In the network shown in Figure 7-3, Routers A and B are the two hub routers in a dual-homed hub-and-spoke topology, whereas Routers C through H are the spoke, or remote, routers. In most hub-and-spoke networks, the remote sites are connected to each hub through a single physical link, virtualized into multiple logical links. The primary issue network engineers normally face with hub-and-spoke networks is simple scale. For instance, there are hub-and-spoke networks deployed today with more than 800 remote sites connected to a single hub site or a pair of hub sites. This is a large number of neighbor relationships for a single router to maintain.

Large-scale hub-and-spoke networks can destabilize an entire network when they have trouble converging, or adjusting to large scale network changes, such as a single link flap causing all the remote neighbors to be reset at the same time. Network engineers should consider a number of rules of thumb when designing a hub-and-spoke network, and then managing the connection of a hub-and-spoke network to a network, including the following:

- Separate the hub-and-spoke network from the rest of the network by making it into a separate failure domain. This is a more specific application of the general rule to divide complexity from complexity where possible.

- Reduce the amount of information being transmitted to the spoke routers to the minimum possible, by using route filters, aggregation, and special protocol capabilities, such as Enhanced Interior Gateway Protocol (EIGRP) stub routers or Open Shortest Path First (OSPF) Protocol link-state advertisement (LSA) filtering.

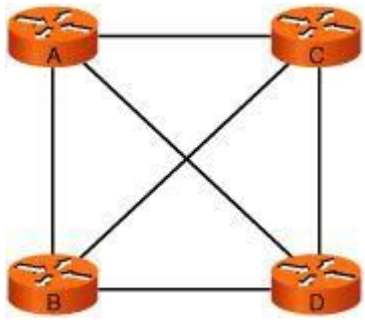
- Reduce the amount of information transmitted to each remote site router. For instance, instead of using fast routing protocol hellos to discover network or link failures, use Layer 2 mechanisms where possible to discover failures.

- Reduce the role the spoke routers play in the convergence of the network where possible.

Ring Topologies

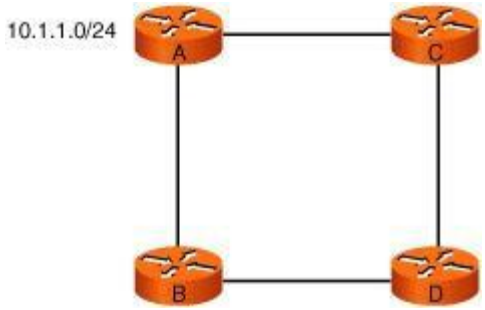
Ring topologies are probably the most common topology, although it sometimes takes a bit of work to recognize a ring. For instance, the network shown in Figure 7-4 has a ring topology embedded in it.

FIGURE 7-4 An Embedded Ring Topology



The ring topology is the set of links around the outside of the four routers shown. The main issue with ring topologies is the process required to converge a ring topology when a single link along the ring fails, or a link to a network outside the network fails, as shown in Figure 7-5.

FIGURE 7-5 A Simple Ring Topology



Assume Router D is using the path through Router C to reach 10.1.1.0/24. What is the convergence process Routers C and D must undertake to converge on the alternate path to 10.1.1.0/24, through Router B? When the link failure has been detected, Router C must recompute its best path, so it now uses Router D to forward traffic to 10.1.1.0/24, rather than the failed link through Router A:

- For EIGRP, this involves a query process that takes, at minimum, 100 ms per hop, and could take much longer. Packets are dropped during this convergence process, because EIGRP prevents routing loops by dropping packets.

- For link-state protocols, this involves the flooding of new link-state information, followed by a shortest path first (SPF) calculation. In most cases, the timing of this flooding and recalculation process will result in a short loop between Routers C and D.

No real solutions prevent these types of problems within ring topologies. The general rule of thumb is to weigh the advantages of a ring topology, in terms of network connectivity at a lower cost, against the disadvantages of routing convergence in such topologies. Cut-through links, which make the ring into a set of triangles, often cut these convergence issues down to a minimum or eliminate them altogether.

Chapter 8. Tunneling Technologies

Although tunneling technologies are generally thought of as being focused in the service provider space, they are actually finding a home in nearly all large-scale networks. This chapter does not focus on how the various available tunneling technologies work, but rather on how they are used, and what impact they can have on a network's design.
Beyond understanding the operation and configuration of any given tunneling technology, a network designer needs to understand some specific factors that will impact the technology's interaction with network design, including the following:

- How does the tunneling mechanism interact with applications running on top of the tunnel? Can packet ordering change within the tunnel, or will the increased overhead of the tunnel headers impact application performance? What are the application's security requirements? What sorts of traffic does the application send, and what can the tunnel carry?

- How well will the tunneling mechanism scale, and how manageable will it be? Will you need to build a full mesh of tunnels to support business or application communications requirements? Will a tunnel supporting a broadcast domain between all sites support the number of sites required?

- How will the tunneling mechanism interact with routing? How can you steer traffic into the tunnels, and steer the tunnels through the network? Will you need to run routing over the tunnels to build reachability through them?

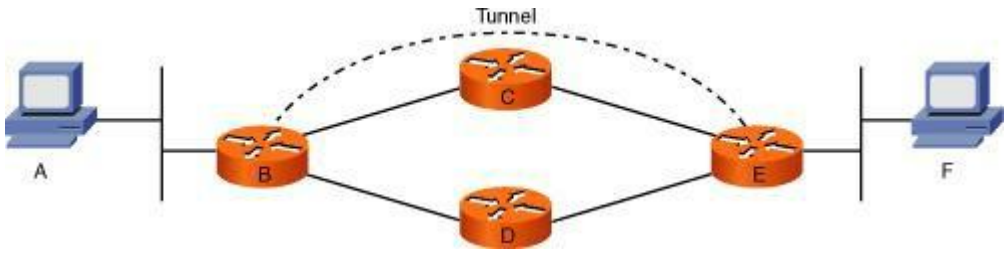
Interaction with Applications

There are a number of ways in which an application will interact with tunneling technologies, including packet-ordering issues, security, failover mechanisms, maximum transmission unit (MTU) issues, and the type of traffic being carried. Each subsection here considers one of these different areas, and points out some of the areas a network designer needs to consider for different technologies.

Packet Ordering

Packets may not leave a tunnel in the same order in which they entered the tunnel. This is no different from the way networks work normally, but the problem can be nonobvious, or harder to resolve, in a tunnel environment, because of the level of indirection between the transport and the routing system, as shown in Figure 8-1.

FIGURE 8-1 Tunnel Packet Reordering



Packets flowing from Host A to Host F, and inserted in the tunnel shown between B and E, may flow over the path through Router C or through the path through Router D. If you examine the traffic from Host A, the traffic will appear to pass through two hops, and yet will arrive at Host D out of order. Unless the network engineer is aware there is a tunnel, and can work with the underlying infrastructure, there is no way to discover the source of this problem, or correct it.

Resolving out-of-order packets flowing through tunnels involves using the same techniques you use to resolve out-of-order packets flowing through any other IP network. The primary culprit for out-of-order packets is load sharing between two equal- or unequal-cost paths. Most network equipment vendors tune their switching paths to load share per flow, rather than per packet, to avoid out-of-order packets.

If per-flow load sharing is used, however, tunneled traffic will typically

take only one of the two available paths. This does prevent out-of-order packets, but it also reduces the effectiveness of load sharing through the network. Some implementations of IP routing (switching) will examine the IP headers within the tunnel header to load share between multiple flows within the tunnel.

Security

There tends to be a misconception within the networking community that placing traffic within a tunnel, no matter what type of tunnel, somehow provides a higher level of security than sending the traffic "raw" through the network. This is particularly true for tunnels that carry tunneled traffic within Layer 2 or "Layer 2.5" frames of some type, such as Virtual Private LAN Service (VPLS) or Layer 2 Tunnel Protocol Version 3 (L2TPv3). This is absolutely not true: Tunneled traffic is no more secure than any other traffic traversing a network.

The only tunnels that provide any sort of security are those using cryptographic mechanisms to protect the traffic passing through the tunnel. In general practice, this would only include IPsec tunnels (although other types of secure tunneling technologies may be available in the future).

The primary concern for a network designer in this area is to pay attention to the security requirements of the application when determining the tunnel type to be used. Any sort of banking transaction, or data that carries information that could be stolen for identity theft or any other white-collar crime, should be carried in an encrypted tunnel, unless the application itself encrypts the data before placing it on the network.

Failover and Recovery

Just as with any other routed IP network, the speed at which the network recovers from a failure may be a critical component of application performance. Tunneled topologies overlaid on top of routed IP networks are no different in this regard. The primary difference when dealing with

tunneled topologies overlaying a routed topology is the interaction between the convergence of the two topologies.
Although it might appear obvious that the tunneled topology cannot converge faster than the routed topology on which it rides, this is not really true in all cases. For instance, in the case of Multiprotocol Label Switching Traffic Engineering (MPLS-TE), specifically, there are fast reroute mechanisms that do not require the underlying network to converge as fast as the overlaying tunnels.
In general, unless the tunneling technology provides for some form of fast convergence beyond the convergence of the underlying network, it is not safe to try to make the tunnel topology converge more quickly than the underlying topology. The result of making the tunnels converge more quickly than the underlying technology will generally be loops or other network problems that could lead to a convergence failure.

MTU Issues

Tunneling technologies always add some amount of overhead in the form of additional headers added to the packet; this is, by definition, what tunneling does. This will inevitably decrease the maximum transmission unit (MTU) size available along a path.
Some applications may have problems dealing with the decreased MTU sizes, or may run, but perform more poorly than they would with a larger MTU. To counter these problems, a network engineer should consider using underlying physical links that allow large MTUs. For instance, Ethernet jumbo frames can allow the tunneling overhead to be taken up in the MTU of the underlying links, rather than in the amount of data that can be transmitted in each packet.
Network engineers should encourage applications to be implemented with path MTU discovery enabled to optimize their performance over tunneled topologies where possible.

Traffic Carrying

Each tunneling technology is designed to carry specific types of frames

or packets. The following table summarizes the types of frames of packets each tunneling type can generally support.

Technology	Frame of Packet Types Supported
MPLS L3VPN	IP packets
MPLS L2VPN	Ethernet frames
VPLS	Ethernet frames
GRE tunnels	A wide variety of frame and packet types, including IP, IPX, and many others
L2TPv3	Ethernet frames

This can impact the ability of a particular tunneling technology to support any given application. If an application requires a Layer 2 link between specific points in the network, the network designer must choose a tunneling technology that can carry Layer 2 frames across the underlying network.

Scaling

The primary factors impacting the scaling of any given tunneling technology is the way in which tunnels are formed. Generally, tunnels will either form point-to-point circuits overlaying the network or they will form a broadcast domain overlaying the underlying network. There are positive and negative aspects in both cases:

- If a tunneling mechanism builds a Layer 2 broadcast domain over the underlying network, it will generally require smaller amounts of configuration per tunnel edge device. The edge device will only need to be configured to connect to a "cloud," and not to every other device within the tunnel overlay network.

- If a tunneling mechanism builds any sort of logical point-to-point link, it will tend to require more configuration at each edge device, but it will provide more control over the number of neighbors any given device on the tunnel topology must maintain state with.

This is no different from the tradeoffs faced when considering a broadcast link infrastructure versus a point-to-point partial- or full-mesh link infrastructure in a routed network. The network designer needs to

pay attention to neighbor counts and state, failure domains, and configuration complexity.
VPLS is currently the only tunneling technology that provides a full-broadcast overlay on top of a routed infrastructure.
RFC 2547 L3VPNs provide a middle alternative to these two options. Although an RFC2547 L3VPN does require a mesh of point-to-point tunnels to build connectivity, it does not require routing adjacencies to be formed over every point-to-point tunnel built. Instead, RFC 2547 L3VPNs inject the overlay topology reachability information into the underlying routing protocol in a way that makes the overlay topology routing system unaware of the tunnel overlay topology. The following section explains this concept more fully.

Interaction with Routing

A network designer needs to consider four primary factors when working with tunnel topologies on top of routed networks:

- How to provide reachability between the tunnel endpoints
- How to provide reachability to destinations reachable through the tunnels
- How to steer traffic into the tunnel topology
- How to engineer the tunneled traffic as it passes through the underlying routed topology

Each of these is covered in one of the following subsections.

Routing Under the Tunnels

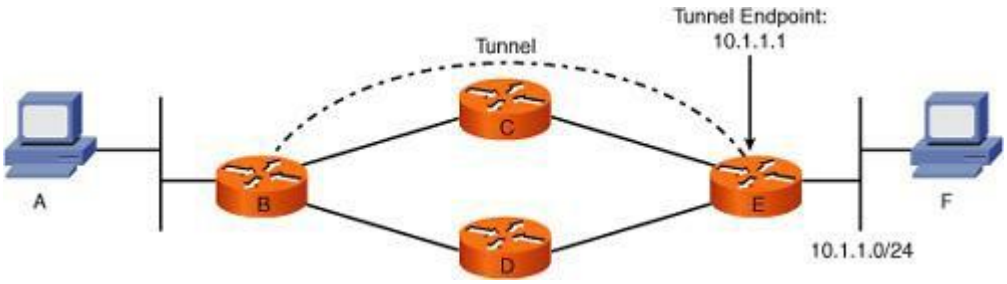
With nearly all tunnel types, the network designer must consider how the device at the tunnel headend knows where the tunnel tailend is and how to reach it. Normally, this is a matter of providing reachability to the IP

address of the tunnel tailend, but not always. In the case of MPLS tunneling, there must be some way of distributing the labels, either through the Label Distribution Protocol (LDP), or through an Interior Gateway Protocol (IGP), such as Intermediate System-to-Intermediate System (IS-IS) Protocol, or through Border Gateway Protocol (BGP).
There are two important considerations for a network designer in the area of providing reachability to the tunnel endpoints:

- Making certain the tunnel endpoints do not pass through the tunnels themselves. Generally, this is accomplished by making certain there is a more specific route to the tunnel endpoint outside the tunnel than is advertised through the tunnel. Figure 8-2 illustrates this concept.

- Making certain the route to the tunnel end point steers the tunnel traffic through the network in a way the designer anticipates and plans for.

FIGURE 8-2 Advertising Tunnel Endpoints



In the network illustrated in Figure 8-2, the tunnel endpoint lies within the destination reachable through the tunnel. If the best path to 10.1.1.1 is through the tunnel itself, the tunnel will collapse into itself, or oscillate between being up and down very rapidly. To prevent this, the 10.1.1.0/24 network should be advertised through the tunnel at a lower cost than through Routers C and D, and a host route, or a route that is more specific than the 10.1.1.0/24 route, should be advertised only through Routers C and D, and not through the tunnel itself.

Routing Through the Tunnels

There must also be some way to provide reachability through the overlay of tunnels. It is easiest to think of the overlay of tunnels as a logical topology sitting on top of the physical topology, and provision routing as if the tunnels overlaying the network were just another physical topology.
There are four ways to handle providing routing through tunnels:

- Running multiple routing protocols, or multiple instances of a single routing protocols, so that the tunnel topology's routing does not intermix with the underlying infrastructure routing

- Running a single routing protocol, but filtering routes and adjusting metrics so that routing information is correctly carried for both the tunnel endpoints and the destinations reachable through the tunnels

- Running multiple protocols, and injecting, or redistributing, the destinations that are reachable through the tunnels into the protocol responsible for providing tunnel endpoint reachability

The first requires careful tuning of metrics and route advertisements. The second requires multiple routing protocols, or multiple routing instances. The third requires some form of redistribution mechanism that can carry the routes through the underlying network without mixing the two types of routing information. This third mechanism is the process used by RFC 2547 L3VPNs.

Steering Traffic into the Tunnels

Two mechanisms are available for steering traffic into a tunnel:

- **Class-based selection**: This mechanism classifies traffic in some way, generally based on the application or data stream type, and pulls traffic within specific classes into the tunnel.

- **Destination-based selection**: This mechanism uses filtering or

metric weighting to ensure only destinations that should be reachable through the tunnel are, actually, reachable through the tunnel.

Traffic Engineering

Traffic engineering, with tunnel topologies, falls into three general categories:

- Using routing protocol metrics to steer the tunnel traffic through the network
- Using routing protocol metrics to steer traffic into specific tunnels overlaying the network
- Using specific traffic engineering technologies particular to the tunneling technology to steer traffic within the tunnel through the network

The first two types of traffic engineering fall within the parameters of normal routed traffic design and engineering. MPLS is almost the only tunneling technology that enables the network designer to engineer traffic through paths not normally chosen by IP through the network. MPLS can use information learned through Resource Reservation Protocol (RSVP), or manually configured information, to restrict the paths the shortest path first algorithm can use to reach a given destination.

Chapter 9. Network Management

Network management is an area often relegated to the final phase of network design, the one called, “I’ll think about it when I get time.” This is not a good approach, however, because the overall health of the network depends on network management.
Network management moves well beyond the concept of tools and techniques, and into building an environment in which anyone working on the network will know why things have been done the way they have, where the network is going, and how the network fits with the business. This chapter focuses on tools and techniques because these are the focus on the CCDE written examination, but a good network engineer will focus on a larger architectural plan.

Available Tools

Tools are an important part of the network management process. The primary problem a network engineer will face in this area is the general discontinuity between implementations and tools. A single tool will not be sufficient to gather the information needed to build a complete picture of the network.
When working with network management tools, the network designer primarily needs to consider which tools can be used for what purposes and apply them correctly. When applying a specific tool, a network designer needs to consider the following:

- Is the tool most appropriate for longer-term monitoring of large amounts of data, or for point monitoring of specific information?

- How is the information gathered, stored, and analyzed? Does the format of the information lend itself to easy analysis by tools? Could the information change format on a regular basis?

- Can the tool lose data in some way? How much information, and under what conditions? What would be the impact of lost information on the accuracy or usefulness of the tool?

Some of the tools and their characteristics are listed here:

- **Console output** (such as the output of **show** commands)

These can sometimes provide the most detailed information available, but this information also tends to be the most difficult to access, interpret, and monitor over time.

Console output is most useful for troubleshooting in real time because it generally provides the most detailed information available on a real-time, interactive basis.

Console output can also prove useful for building a network baseline. Most console output does not provide any useful information unless you know what the output should look like. For instance, Enhanced Interior Gateway Protocol (EIGRP) active routes might appear to be part of a network convergence problem; there's no way to tell unless you know whether EIGRP routes are normally active in the network, how long they remain active, and so on.

- **Various forms of logging** (such as event logs, console logs, and others)

These outputs can give you real time information about specific events, and the sequence of specific events, within the network.

Logging output can be difficult to gather, manage, and monitor. Each platform, and even each component or application on each platform, can have different log formats, and these log formats can change over time.

Logging output can be captured from most systems via syslog, which is a TCP-based protocol.

Logging output can prove useful, at low volumes, to keep track of specific network events and network state. Logging is not useful, for instance, for keeping track of traffic flows, or highly dynamic events within the network, but it is useful for understanding the flow of events when a routing protocol fails to converge.

Logged output can be dropped in multiple ways, because of buffer overflows, rate limiting of log messages on the platform being monitored, and other problems.

• **Simple Network Management Protocol (SNMP)**

This is a standardized protocol used to access different parameters and statistics on network devices. Several versions of SNMP are available (as covered in the "Management Protocols" section). Generally, SNMP can access a wide array of variables, and can generate traps when the states of particular variables change.

SNMP cannot generally access all the same information as other mechanisms, such as console output and logs.

SNMP can drop information if the network or the device being queried, or sending the trap, is heavily loaded. SNMP is usually implemented using User Datagram Protocol (UDP) as the underlying protocol.

SNMPv2 is generally considered unsecure because the only security mechanism available is a plain-text community string. SNMPv3 is considered more secure than SNMPv2, but is less widely deployed.

• **Access control mechanisms**

Access control mechanisms are generally designed to control traffic flow through the network, including the ability to allow or deny traffic of certain classes, from specific sources, or to specific destinations, from passing through the network. Quite often, these access control mechanisms can keep track of how many packets

have been allowed or denied. By creating an access control that permits all packets, but allows specific flows separately from general traffic flowing through the network, a network engineer can determine how many packets within a specific flow have passed through given devices within the network.

These mechanisms can be combined with other mechanisms, such as logging and syslog, to gain an understanding of a specific flow of traffic through the network.

These mechanisms are not useful for general network traffic-flow monitoring because you need to know what you are looking for before you set the filters up. On the other hand, they can prove very useful for understanding the size, path, and characteristics of specific flows within a network.

- **NetFlow**

NetFlow is a mechanism designed to sample the traffic flows within a network, or at the edge of a network. The general concept is that information about a subset of the flows within a network is exported to a monitoring device. This information is later analyzed, generally using a statistical process, to understand traffic flows within the network.

NetFlow is often used as part of a billing mechanism within a network, and to track general traffic patterns within the network. Often, this information is used to find problems in and adjust traffic engineering.

Management Access

One area of importance to consider is whether in-band network management is sufficient, or whether out-of-band mechanisms need to be considered. Two things need to be considered in making this decision:

- Is it important for network devices to be accessible during network outages? If a specific network device must be accessible no matter what the current network conditions are, out-of-band management should be considered for that device.

- Do network security requirements infer that some network management tasks or interfaces should not be accessible to the users of the network? If so, out-of-band network management should be considered.

Out-of-band management can be partially deployed within a network. For instance, a particular data center may be provided with out-of-band management, to provide access if all the standard links into the data center fail, rather than providing out-of-band access to all the devices within the data center.

Auditable Factors

Auditable factors include any element of the network characteristic that could be monitored for regulatory reasons. Examples of these factors include the following:

- The security level of specific transactions within the network, such as credit card transactions, banking transfers, and others, might be included in the information a network manager needs to track.

- The processing of specific actions within the network might be verified or recorded through network management mechanisms. For instance, a backup that is supposed to run once a night might be verified using application mechanisms, and cross-verified by tracking traffic flows within the network.

Network engineers should consider the regulatory environment when designing network management systems, and choose the correct tools to monitor and record the information requirements. In these applications,

the network engineer needs to carefully consider possible loss of management data.

Configuration Management Tools and Practices

There are a large variety of network configuration management systems available. Network engineers should consider these systems, and use them where they make sense. Some of the positive and negative aspects of such systems include the following:

- Network configuration management can encourage consistency in configurations throughout the network.
- Network configuration management can encourage complex configurations that are difficult to troubleshoot.
- Network configuration management can support a strong change control management system, reducing human errors generated in on-the-fly changes.
- Network configuration management can encourage less than optimal configurations, such as long access lists as opposed to shorter prefix lists, because the difficulty of generating long lists of information is simplified.
- Network configuration management can encourage better review procedures, and provide better tracking for configuration changes.

Chapter 10. IP Quality of Service

Many IP networks today carry traffic from a wide range of applications requiring different levels of performance from the network for optimal operations. It is important that traffic from mission-critical applications be given priority over ordinary traffic, and at the same time delay and jitter be minimized for any time-sensitive multimedia and voice applications.
Quality of service (QoS) refers to the network's capability to control resources and provide the appropriate level of service for selected traffic over various underlying technologies. Because of the convergence of voice, video and data, QoS is becoming an increasingly critical element in network design considerations and deployment of IP-based networks.
The Internet Engineering Task Force (IETF) has proposed two primary architectures: integrated services (IntServ) and differentiated services (DiffServ).

Integrated Services

The IntServ model is a flow-based model with a signaling component to reserve adequate resources. The IntServ model relies on the Resource Reservation Protocol (RSVP) to signal and reserve the desired resources for each flow (source IP and port number, destination IP and port number, and the transport protocol) in the network. The signaling protocol guarantees that adequate resources are available at each RSVP hop for the flow before admitting the flow onto the network.
Two primary services can be requested via RSVP for a given flow.

- **Guaranteed services (GS)**: It is a strict guaranteed service that provides for firm bounds on end-to-end delay and assured

bandwidth for traffic that conforms to the reservation specifications.

- **Controlled load service (CLS)**: It is a service that approximates the end-to-end performance of the reserved flow as if it were a best-effort flow on the same network with unloaded conditions. The end-to-end performance is maintained even in loaded conditions using various methods, including admission control.

Every RSVP node along a flow's path must be capable of signaling and refreshing the required QoS for each flow. IntServ is problematic when scaling to large networks because of the large number of flows requiring each node to maintain the state of all flows traversing that node.

IntServ uses a flow-based approach where a signaling protocol is used along the same path as the data path for a given flow. The signaling protocol guarantees that adequate resources are available (at each hop) for the flow before admitting the flow onto the network.

Consider a host H1 that needs to send data to host H2 with certain bandwidth and delay guarantees, as shown in Figure 10-1. The following steps would have to occur in the stated order:

1. H1 sends an RSVP PATH message that contains the flow's requirements along the data path from H1 to H2.

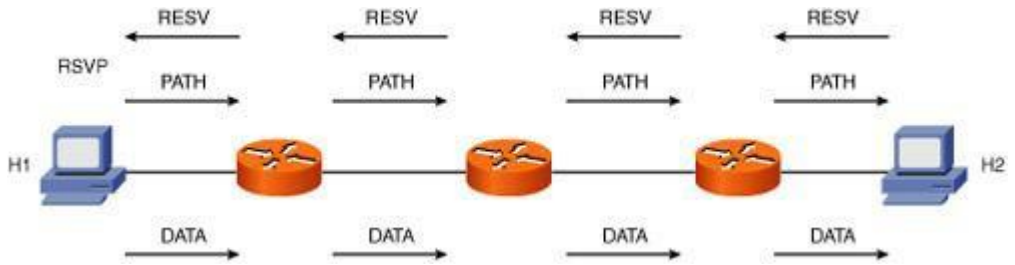
2. The RSVP PATH message is intercepted at each node along the path for processing.

3. Once it reaches H2, H2 sends a RSVP RESV message back to H1 along the same path.

4. The RSVP RESV message is intercepted at each RSVP node along the path for processing and confirmation of resource reservation.

5. Once H1 receives the RSVP RESV message, H1 will start sending the data toward H2.

FIGURE 10-1 RSVP Operations



Differentiated Services

DiffServ specifications define a number of tools that could be used to build a variety of services. It uses the concept of defining a small number of service classes. A service class could be defined as a set of traffic requiring specific delay, jitter, and loss across the network. All the traffic entering a DiffServ network or domain is classified into one of the defined service classes on a packet-by-packet basis. Each service class receives a particular forwarding treatment (per-hop behavior, PHB) associated with that class at each and every hop.

When a packet enters the network, it is classified based on any of the fields in the IP packet. Classification may also include non-IP based attributes (for example, ingress port, ATM virtual circuit [VC], and Ethernet VLAN). The incoming traffic may be metered, and based on policies it may get dropped, marked, or shaped. This treatment of traffic at the ingress of the network is called *traffic conditioning*. Traffic transiting through a DiffServ network may experience traffic conditioning while exiting the network, too.

The packet entering the DiffServ network is marked with associated differentiated services code point (DSCP) in the Differentiated Services Field (DSField). This scheme scales well to very large networks, because the routers have to maintain the state of just a few service classes as opposed to a large number of flows. In case of IPv4, the DSField is the 6 most significant bits of the IP Type of Service (ToS) byte, and in case of IPv6 it is the 6 most significant bits of the Traffic Class byte. Value set or encoded in the DSField is called the DSCP. In legacy networks, 3 bits of the DSField are used and are called IP

Precedence. To help maintain backward compatibility with IP Precedence, the DiffServ standard defines a Class Selector code point field within DSCP to create different levels of classes.
Once classification and marking is done, all the DiffServ routers must use the DSCP value to select a PHB for a given packet. PHB defines the forwarding treatment a given packet will receive based on the service class encoded in the DSCP. The forwarding treatment is defined by the allocation of resources per class and packet-drop policies.
A few fundamental PHBs are defined in DiffServ specifications, Expedited Forwarding (EF), Assured Forwarding (AF), Default Forwarding (DF) and Class Selector (CS):

- **Expedited Forwarding (EF)**: Expedited Forwarding is used for traffic requiring low delay, low loss, and low jitter. Voice and video applications are good candidates for EF PHB. The traffic is usually rate limited to protect the network from potential starvation of other service classes. This PHB is usually implemented using low-latency queuing or some sort of priority queue.

- **Assured Forwarding (AF)**: Assured Forwarding is used for traffic requiring different levels of assurances for delay, loss and jitter. It is suitable for TCP-based applications. Multiple AF behaviors can be defined, and each associated service class is serviced in its own queue, enabling independent bandwidth management for different AF service classes.

- **Default Forwarding (DF)**: Default Forwarding is generally used for service classes that require best-effort treatment from the network. In most of the cases, some bandwidth is assigned for this PHB.

- **Class Selector (CS)**: Class Sector is used to provide backward compatibility for the code points and PHBs that are widely used in the networks today. One Class Selector is Network Control (CS6), which was defined pre-DiffServ and is still in use today.

Table 10-1 shows the suggested service classes and their corresponding

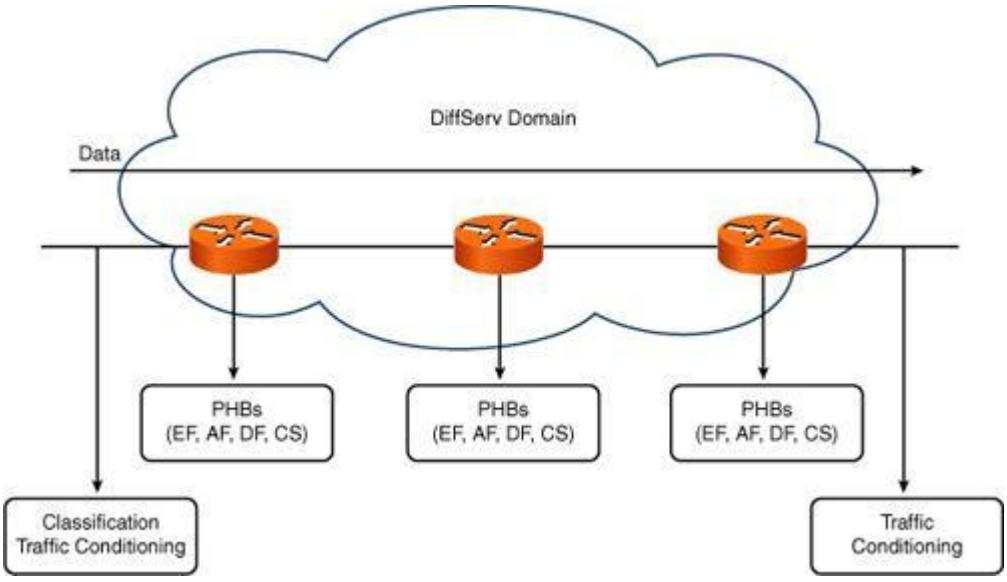
DSCP values.

Table 10-1 Service Classes, DSCP Names, and Values (RFC 4594)

Service Class	DSCP Name	DSCP
Network Control	CS6	110000
Telephony	EF	101110
Signaling	CS5	101000
Multimedia Conferencing	AF41 AF42 AF43	100010 100100 100110
Real-Time Interactive	CS4	100000
Multimedia Streaming	AF31 AF32 AF33	011010 011100 011110
Broadcast Video	CS3	011000
Low-Latency Data	AF21 AF22 AF23	010010 010100 010110
OAM	CS2	010000
High-Throughput Data	AF11 AF12 AF13	001010 001100 001110
Standard	DF	000000
Low-Priority Data	CS1	001000

Figure 10-2 shows a typical DiffServ domain with appropriate functions in the network.

FIGURE 10-2 DiffServ Domain



Tools for QoS

Many tools are used to implement QoS in a network, including policers, shapers, weighted random early detection (WRED), and class-based weighted fair queuing (CBWFQ).

Policers are set to police the rate of the selected traffic. The traffic that is above the specified rate is classified as nonconformant, and the traffic that is within the specified rate is classified as conformant. Different actions could be taken based on a policer's classification of packets, including marking, dropping, or transmitting. Policers are widely used to limit the amount of traffic admitted to the priority queue. TCP-based applications generally do not react well to a policer that drops packets exceeding the specified rate because this is seen as tail drop. Marking conformant and nonconformant traffic differently may lead to out-of-order packets if the two markings are mapped to different queues.

Shapers measure the rate of the selected traffic, but unlike policers they buffer the excess traffic. Shapers do not drop or mark any packets. Shapers may introduce delay and jitter, because of buffering of excess traffic, and therefore are not suitable for time-sensitive applications.

WRED is a congestion-avoidance mechanism. WRED monitors queue depth and starts dropping packets before the queue gets full and starts tail dropping. WRED takes into account the weight (DSCP or IP Precedence) and the queue depth in making packet-drop decisions. This tool is designed for applications that have a flow-control mechanism (TCP based) to detect packet drops and adapt to the available capacity. WRED is not appropriate for real-time applications.

CBWFQ is a queuing mechanism where appropriate resources are assigned to various queues belonging to different service classes. The queues are served in a round-robin fashion (with the exception of a low-latency or priority queue). Packets in the priority queue are given higher priority than other queues. Bandwidth of the interface could be carved out and assigned to each individual queue. More than one service class could be mapped to a single queue. It is not a good idea to put widely different service classes into the same queue.

IP QoS and Layer 2

Networks that have to transit multiple underlying Layer 2 technologies have to make sure that the IP traffic receives appropriate treatment when traversing Layer 2 networks to maintain end-to-end service definition. In point-to-point dedicated connections (for example, dark fiber or back-to-back GigE connection between two nodes), this is not a concern. The QoS design needs consideration when media such as ATM, Frame Relay, and switched Ethernet are present.

- **ATM**: ATM carries IP traffic in the form of cells across the network. An ATM cell header has only 1 bit, CLP (Cell Loss Priority), to indicate the priority of the cell. Multiple service classes are mapped to each of the two service classes available in ATM. Another approach that has been taken is to have multiple ATM VCs among the routers with appropriate bandwidth assignment, and then map each VC to a corresponding DSCP value. When a packet is received by a router, a particular VC is used for transmission, from a set of VCs to the same destination, based on the DSCP value.

- **Frame Relay**: Frame Relay is quite similar to ATM; it has only 1 bit, DE (Discard Eligible), to notify the Frame Relay switches whether a given packet is high priority. Similar mapping would be required from IP DSCP to Frame Relay DE bit.

- **Ethernet**: Routers cannot mark the frames for QoS if they are connected using nontrunk Ethernet to the switched Ethernet domain. There is no option at Layer 2 in a nontrunk Ethernet to indicate a class of service. The QoS design would have to allocate resources in the Layer 2 domain based on ports, VLANs, MAC addresses, and so forth. The other option is to have the Layer 2 switches read the IP DSCP markings and have the switches treat the packets based on IP DSCP.

If a .1Q trunk is used, the routers have the capability to mark the .1p bits as the packet is sent. All the Ethernet switches will notice the .1p bits

setting and will provide appropriate treatment based on policies defined on the switches.

IP QoS and MPLS

Multiprotocol Label Switching (MPLS) is a tunneling technology that uses LSPs (label-switched paths) to transport packets across the network. LSPs are set up by exchanging labels among routers (via label-distribution protocols). After the packet has been encapsulated in MPLS, the underlying IP header is not visible to many label-switched routers (LSRs). In an MPLS header (shim), there are 3 experimental (exp) bits that are used for marking. Based on the value set in the exp bit, all the LSRs provide appropriate treatment to MPLS packets.
There are primarily two kinds of LSPs from a QoS perspective: E-LSPs (Exp-inferred-class LSP) and L-LSPs (Label-inferred-class LSP).

- **E-LSP**: Carries traffic of multiple classes of service, and these classes are identified by exp value. The label value is used only for label switching, and exp is used for QoS class identification. In this case, PHB is determined by exp only.

- **L-LSP**: L-LSP carries traffic from only one class of service. If multiple classes of service are being used, multiple LSPs are set up to the same destination. When L-LSPs are signaled, the class of service is also signaled. When the LSRs receive a packet on an L-LSP, they know the class from the label and use the exp to determine the drop probability. So, PHB is determined based on label and exp.

MPLS Traffic Engineering (MPLS-TE) uses RSVP extensions to signal an LSP and propagate label information along the path. MPLS-TE can set up LSPs to destinations along a path other than the shortest Interior Gateway Protocol (IGP) path, and bandwidth reservations can also be made. To set up LSPs with appropriate resources, MPLS-TE has to have a full view of the topology and resources available. To have a full

topological view with resource information, extensions to Open Shortest Path First (OSPF) Protocol or Intermediate System-to-Intermediate System (IS-IS) Protocol are used. While a path is being computed, the elements that do not have adequate resources are pruned, and shortest path among the nodes that have appropriate resources is computed.
DiffServ-aware MPLS-TE is a way of dividing up bandwidth resources into different class pools. When MPLS-TE is signaled via RSVP, you can specify which pool to use to reserve the bandwidth. This provides a way to break up networkwide resources into different classes and then do admission control and reservation based on a per-class basis.
MPLS TE also provides the capability of nailing up backup LSPs. If the primary LSP fails, the LSR detecting the failure uses the backup LSP right away, and no signaling is required to set up a new LSP, resulting in fast rerouting. Backup LSPs can be set up for protecting links, nodes, or end-to-end paths. The protection schemes cannot protect against the failure of headend or the tailend LSR.

QoS Performance and Measurement

It is important to be able to objectively measure QoS received by an application (that is, use a standards-based method). The IP Performance Metrics (IPPM) workgroup within the IETF has been tasked to define standard metrics related to quality, performance, and reliability of IP transport. Here is a list of some of the most commonly used metrics to measure the QoS received by an application or service class from the network:

- Bandwidth
- One-way delay
- Round-trip delay
- Jitter

- Packet loss
- Packet reordering

In many of the deployments, active monitoring is used, where probes are sent across the network on a periodic basis to measure network performance. These metrics are usually used to measure the performance of the transport but not necessarily application-specific performance. For some of the applications, special algorithms have been developed to measure application-specific performance.

IntServ RFCs

RFC 1633, *Integrated Services in the Internet Architecture: An Overview*
RFC 2211, *Specification of the Controlled-Load Network Element Service*
RFC 2212, *Specification of Guaranteed Quality of Service*
RFC 2215, *General Characterization Parameters for Integrated Service Network Elements*
RFC 2205, *Resource ReSerVation Protocol (RSVP)*

DiffServ RFCs

RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*
RFC 2475, *An Architecture for Differentiated Services*
RFC 2597, *Assured Forwarding PHB Group*
RFC 3140, *Per Hop Behavior Identification Codes (Obsoletes RFC 2836)*
RFC 3246, *An Expedited Forwarding PHB (Obsoletes RFC 2598)*
RFC 4594, *Configuration Guidelines for DiffServ Service Classes*