CISCO

CCNP SWITCH 642-813

Quick Reference

Denise Donohue

Cisco Press

CISCO

CCNP SWITCH 642-813 Quick Reference

Denise Donohue

About the Author

Denise Donohue, CCIE No. 9566, is a senior solutions architect for ePlus Technology. She consults with companies to design updates or additions to their data and VOIP networks. Prior to this role, she was a systems engineer for the data consulting arm of SBC/AT&T. Denise has been a Cisco instructor and course director for Global Knowledge and did network consulting for many years. Her CCIE is in Routing and Switching.

About the Technical Editor

'Rhette (Margaret) Marsh has been working in the networking and security industry for more than ten years, and has extensive experience with internetwork design, IPv6, forensics, and greyhat work. She currently is a design consultant for Cisco in San Jose, CA, and works primarily with the Department of Defense and contractors. Prior to this, she worked extensively both in the financial industry as a routing and switching and design/security consultant and also in an attack attribution and forensics context. She currently holds a CCIE in Routing and Switching (No. 17476), CCNP, CCDP, CCNA, CCDA, CISSP and is working towards her Security and Design CCIEs. In her copious free time, she enjoys number theory, arcane literature, cycling, hiking in the redwoods, sea kayaking, and her mellow cat, Lexx.

CCNP SWITCH 642-813 Quick Reference   by Denise Donohue

Icons Used
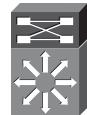
| Router | Route/Switch Processor | Multilayer Switch | Workgroup Switch | PC |

Chapter 3
Spanning Tree

Ethernet network design balances two separate imperatives. First, Ethernet has no capacity for detecting circular paths. If such paths exist, traffic loops around and accumulates until new traffic is shut out. (This is called a broadcast storm.) Second, having secondary paths is good preparation for inevitable link failure.

Spanning Tree is a protocol that prevents loop formation by detecting redundant links and disabling them until needed. Designers can therefore build redundant links, and the protocol enables one to pass traffic and keep the other in reserve. When the active link fails, the secondary link is enabled quickly.

## Understanding the Spanning Tree Protocol

Switches either forward or filter Layer 2 frames. The way they make the forwarding/filtering decision can lead to loops in a network with redundant links. Spanning Tree is a protocol that detects potential loops and breaks them.

A Layer 2 switch is functionally the same thing as a transparent bridge. Transparent bridges:

- Learn MAC (Media Access Control) addresses by looking at the source address of incoming frames. They build a table mapping MAC address to port number.

- Forward broadcasts and multicasts out all ports except the one in which they came. (This is called flooding.)

- Forward unknown unicasts out all ports except the one in which they came. An unknown unicast is a message bound for a unicast MAC address that is not in the switch's table of addresses and ports.

- Do not make any changes to the frames as they forward them.

Spanning Tree

Spanning Tree Protocol (STP) works by selecting a root bridge and then selecting one loop-free path from the root bridge to every other switch. (STP uses the term *bridge* because it was written before there were switches.) Consider the following switched network (see Figure 3-1).

FIGURE 3-1
Example Switched
Topology
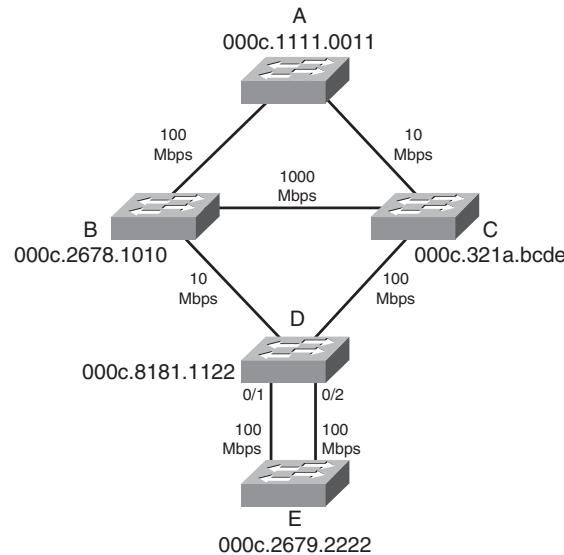
Spanning Tree must select

■ One root bridge

■ One root port per nonroot bridge

■ One designated port per network segment

Spanning Tree Election Criteria

Spanning Tree builds paths out from a central point along the fastest available links. It selects paths according to the following criteria:

- Lowest root bridge ID (BID)

- Lowest path cost to the root

- Lowest sender bridge ID

- Lowest sender port ID (PID)

When reading the path selection criteria, remember the following:

- **Bridge ID:** Bridge priority: Bridge MAC address.

- **Bridge priority:** 2-btye value, 0–65,535 (0–0xFFFF).

- **Default priority:** 32,768 (0x8000).

- **Port ID:** Port priority: port number.

- **Port priority:** A 6-bit value, 0–63, default is 32.

- **Path cost:** This is the cumulative value of the cost of each link between the bridge and the root. Cost values were updated in 2000, and you should see only new cost values, but both are given in the following table (see Table 3-1). Old and new switches work together.

Table 3-1   Spanning Tree Costs

Link Speed	Previous IEEE Specification	Current IEEE Specification
10 Mb/s	100	100
100 Mb/s	10	19
1 Gbps	1	4
10 Gbps	1	2

The STP Election

Spanning Tree builds paths out from a starting point, the "root" of the tree. The first step in selecting paths is to identify this root device. Then each device selects its best path back to the root, according to the criteria laid out in the previous sections (lowest root BID, lowest cost, lowest advertising BID, lowest port ID).

Root Bridge Election

Looking at Figure 3-1, first select the root bridge. Assume each switch uses the default priority.

- Switch A BID = 80–00–00–0c-11–11–00–11

- Switch B BID = 80–00–00–0c–26–78–10–10

- Switch C BID = 80–00–00–0c-32–1a-bc-de

- Switch D BID = 80–00–00–0c-81–81–11–22

- Switch E BID = 80–00–00–0c–26–79–22–22

Switch A has the lowest BID, so it is the root. Each nonroot switch must now select a root port.

Root Port Election

The root port is the port that leads back to the root. Continuing with Figure 3-1, when A is acknowledged as the root, the remaining bridges sort out their lowest cost path back to the A:

- **Switch B:** Uses the link to A with a cost of 19 (link speed of 100 Mb/s).

- **Switch C:** The connected link has a cost of 100 (Ethernet), the link through B has a path cost of 38 (two 100-Mb/s links), and so B is chosen.

- **Switch D:** The link through B has a path cost of 119, the path cost through C to A is 119, the path through C then B is 57, so C is chosen.

- **Switch E:** The lowest path cost is the same for both ports (76 through D to C to B to A). Next check sender BID— sender for both ports is D so that it does not break the tie. Next check sender Port ID. Assuming default port priority, the PID for 0/1 is lower than the PID for 0/2, so the port on the left is the root port.

Designated Port Election

Designated ports are ports that lead away from the root. Obviously, all ports on the root bridge are designated ports (A–B and A–C in Figure 3-1).

- **Segment B–D:** B has the lowest path cost to root (19 versus 119), so it is designated for this segment.

- **Segment C–D:** C has the lowest path cost to the root (100 versus 119), so it is designated for this segment.

- **Segment B–C:** B has the lowest path cost to the root (19 versus 100), so it is designated for this segment.

- **Both segments D–E:** D has the lowest cost to the root (57 versus 76), so it is designated for both segments.

Now the looped topology has been turned into a tree with A at the root. Notice that there are no more redundant links.

FIGURE 3-2
The Active Topology
After Spanning Tree
Is Complete
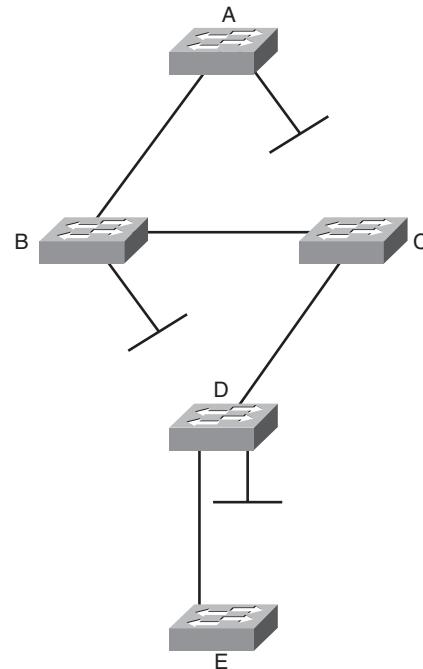
Bridge Protocol Data Units

Switches exchange Bridge Protocol Data Units (BPDU). The two types of BPDUs are Configuration and Topology Change Notification(TCN). Configuration BPDUs are sent every two seconds from the root toward the downstream switches. They:

- Are used during an election

- Maintain connectivity between switches

- Send timer information from the root

TCN BPDUs are sent by a downstream switch toward the root when:

■ There is a link failure.

■ A port starts forwarding, and there is already a designated port.

■ The switch receives a TCN from a neighbor.

When a switch receives a TCN BPDU, it acknowledges that with a configuration BPDU that has the TCN Acknowledgment bit set.

When the root bridge receives a TCN, it starts sending configuration BPDUs with the TCN bit set for a period of time equal to max age plus forward delay. Switches that receive this change their MAC table aging time to the Forward Delay time, causing MAC addresses to age faster. The topology change also causes an election of the root bridge, root ports, and designated ports.

Some of the fields in the BPDU include:

■ **Root bridge ID:** The BID of the current root

■ **Sender's root path cost:** The cost to the root

■ **Sender's bridge ID:** Sender's priority concatenated to MAC

■ **Sender's port ID:** The port number, transmitted as final tie-breaker

■ **Hello time:** Two seconds by default

■ **Forward Delay:** Fifteen seconds by default

■ **Max Age:** Twenty seconds by default

Spanning Tree Port States

When a port is first activated, it transitions through the following stages shown in Table 3-2.

Table 3-2   Spanning Tree Port States

Port State	Timer	Action
Blocking	Max Age (20 sec)	Discards frames, does not learn MAC addresses, receives BPDUs
Listening	Forward Delay (15 sec)	Discards frames, does not learn MAC addresses, receives BPDUs to determine its role in the network
Learning	Forward Delay (15 sec)	Discards frames, does learn MAC addresses, receives and transmits BPDUs
Forwarding		Accepts frames, learns MAC addresses, receives and transmits BPDUs

Per-VLAN Spanning-Tree

The IEEE's version of STP assumes one common Spanning-tree instance (and thus one root bridge) regardless of how many VLANs are configured. With the Cisco Per-VLAN Spanning-Tree (PVST+) there is a different instance of STP for each VLAN. To derive the VLAN BID, the switch picks a different MAC address from its base pool for each VLAN. Each VLAN has its own root bridge, root port, and so on. You can configure these so that data flow is optimized, and traffic load is balanced among the switches by configuring different root bridges for groups of VLANs.

PVST+ is enabled by default on Cisco switches.

Configuring Spanning Tree

To change the STP priority value, use the following:

```
Switch (config)# spanning-tree vlan vlan_no. priority value
```

To configure a switch as root without manually changing priority values, use the following:

```
Switch (config)# spanning-tree vlan vlan_no. root {primary ¦ secondary}
```

To change the STP port cost for an access port, use the following:

```
Switch(config-if)# spanning-tree cost value
```

To change the STP port cost for a VLAN on a trunk port, use the following:

```
Switch(config-if)# spanning-tree vlan vlan_no. cost value
```

To display STP information for a VLAN, use the following:

```
Switch# show spanning-tree vlan vlan_no.
```

To display the STP information for an interface, use the following:

```
Switch # show spanning-tree interface interface_no. [detail]
```

To verify STP timers, use the following:

```
Switch # show spanning-tree bridge brief
```

Portfast

Portfast is a Cisco-proprietary enhancement to Spanning Tree that helps speed up network convergence. It is for access (user) ports only. Portfast causes the port to transition directly to forwarding, bypassing the other STP states. Connecting a switch to a Portfast port can cause loops to develop. Configure Portfast on an interface or interface range:

```
(config-if)# spanning-tree portfast
```

It can also be configured **globally:**

```
(config)# spanning-tree portfast default
```

Rapid Spanning Tree

Rapid Spanning Tree (RSTP) 802.1w is a standards-based, nonproprietary way of speeding STP convergence. Switch ports exchange an explicit handshake when they transition to forwarding. RSTP describes different port states than regular STP, as shown in Table 3-3.

Table 3-3   Comparing 802.1d and 802.1w Port States

STP Port State	Equivalent RSTP Port State
Disabled	Discarding
Blocking	Discarding
Listening	Discarding
Learning	Learning
Forwarding	Forwarding

RSTP Port Roles

RSTP also defines different Spanning Tree roles for ports:

- **Root port:** The best path to the root (same as STP)

- **Designated port:** Same role as with STP

- **Alternate port:** A backup to the root port

- **Backup port:** A backup to the designated port

- **Disabled port:** Not used in the Spanning Tree

- **Edge port:** Connected only to an end user

BPDU Differences in RSTP

In regular STP, BPDUs are originated by the root and relayed by each switch. In RSTP, each switch originates BPDUs, whether or not it receives a BPDU on its root port. All eight bits of the BPDU type field are used by RSTP. The TC and TC Ack bits are still used. The other six bits specify the port's role and its RSTP state and are used in the port handshake. The RSTP BPDU is set to Type 2, Version 2. PVST is done by Rapid PVST+ on Catalyst switches.

RSTP Fast Convergence

The Rapid Spanning Tree process understands and incorporates topology changes much quicker than the previous version:

- **RSTP uses a mechanism similar to BackboneFast:** When an inferior BPDU is received, the switch accepts it. If the switch has another path to the root, it uses that and informs its downstream switch of the alternative path.

- **Edge ports work the same as Portfast ports:** They automatically transition directly to forwarding.

■ **Link type:** If you connect two switches through a point-to-point link and the local port becomes a designated port, it exchanges a handshake with the other port to quickly transition to forwarding. Full-duplex links are assumed to be point-to-point; half-duplex links are assumed to be shared.

■ **Backup and alternate ports:** Ports that can transition to forwarding when no BPDUs are received from a neighbor switch (similar to UplinkFast).

If an RSTP switch detects a topology change, it sets a TC timer to twice the hello time and sets the TC bit on all BPDUs sent out its designated and root ports until the timer expires. It also clears the MAC addresses learned on these ports. Only changes to the status of non-Edge ports cause a TC notification.

If an RSTP switch receives a TC BPDU, it clears the MAC addresses on that port and sets the TC bit on all BPDUs sent out its designated and root ports until the TC timer expires. Enable and verify Rapid STP with the commands:

```
Switch(config)# spanning-tree mode rapid-pvst
Switch# show spanning-tree
```

A version of PVST+ is used with Rapid Spanning Tree, called Per-VLAN Rapid Spanning Tree (PVRST+). You should still configure root and secondary root bridges for each VLAN when using RSTP.

Multiple Spanning Tree

With Multiple Spanning Tree (MST), you can group VLANs and run one instance of Spanning Tree for a group of VLANs. This cuts down on the number of root bridges, root ports, designated ports, and BPDUs in your network. Switches in the same MST Region share the same configuration and VLAN mappings. Configure and verify MST with these commands:

```
(config)# spanning-tree mode mst
(config)# spanning-tree mst configuration
(config-mst)# name region_name
```

```
(config-mst)# revision number
(config-mst)# instance number vlan vlan_range
(config-mst)# end
# show spanning-tree mst
```

To be compatible with 802.1Q trunking, which has one common Spanning Tree (CST) for all VLANs, MST runs one instance of an Internal Spanning Tree (IST). The IST appears as one bridge to a CST area and is MST instance number 0. The original MST Spanning Trees (called M-Trees) are active only in the region; they combine at the edge of the CST area to form one.

Spanning Tree Stability Mechanisms

Spanning Tree has several additional tools for tuning STP to protect the network and keep it operating properly. They include:

- PortFast (discussed previously)
- UplinkFast
- BackboneFast
- BPDU Guard
- BPDU Filtering
- Root Guard
- UDLD
- Loop Guard

UplinkFast

UplinkFast is for speeding convergence when a direct link to an upstream switch fails. The switch identifies backup ports for the root port. (These are called an uplink group.) If the root port fails, one of the ports in the uplink group is unblocked and transitions immediately to forwarding; it bypasses the listening and learning stages. It should be used in wiring closet switches with at least one blocked port.

The command to enable uplinkfast is shown next. Please note that uplinkfast is enabled globally, so the command affects all ports and all VLANs.

```
(config)# spanning-tree uplinkfast
```

BackboneFast

BackboneFast is used for speeding convergence when a link fails that is not directly connected to the switch. It helps the switch detect indirect failures. If a switch running BackboneFast receives an inferior BPDU from its designated bridge, it knows a link on the path to the root has failed. (An inferior BPDU is one that lists the same switch for the root bridge and designated bridge.)

The switch then tries to find an alternate path to the root by sending a Root Link Query (RLQ) frame out all alternate ports. The root then responds with an RLQ response, and the port receiving this response can transition to forwarding. Alternate ports are determined in this way:

- If the inferior BPDU was received on a blocked port, the root port and any other blocked ports are considered alternates.

- If the inferior BPDU was received on the root port, all blocked ports are considered alternates.

- If the inferior BPDU was received on the root port and there are no blocked ports, the switch assumes it has lost connectivity with the root and advertises itself as root.

Configure this command on all switches in the network:

```
(config)# spanning-tree backbonefast
```

BPDU Guard

BPDU Guard prevents loops if another switch is attached to a Portfast port. When BPDU Guard is enabled on an interface, it is put into an error-disabled state (basically, shut down) if a BPDU is received on the interface. It can be enabled at either global config mode—in which case it affects all Portfast interfaces—or at interface mode. Portfast does not need to be enabled for it to be configured at a specific interface. The following configuration example shows BPDU guard being enabled and verified.

```
(config)# spanning-tree portfast bpduguard default
(config-if)# spanning-tree bpduguard enable
# show spanning-tree summary totals
```

BPDU Filtering

BPDU filtering is another way of preventing loops in the network. It also can be enabled either globally or at the interface and functions differently at each. In global config, if a Portfast interface receives any BPDUs, it is taken out of Portfast status. At interface config mode, it prevents the port from sending or receiving BPDUs. The commands are:

```
(config)# spanning-tree portfast bpdufilter default
(config-if)# spanning-tree bpdufilter enable
```

Root Guard

Root Guard is meant to prevent the wrong switch from becoming the Spanning Tree root. It is enabled on ports other than the root port and on switches other than the root. If a Root Guard port receives a BPDU that might cause it to become a root port, the port is put into "root-inconsistent" state and does not pass traffic through it. If the port stops receiving these BPDUs, it automatically reenables itself. To enable and verify Root Guard use the following commands:

```
(config-if)# spanning-tree guard root
# show spanning-tree inconsistentports
```

Unidirectional Link Detection

A switch notices when a physical connection is broken by the absence of Layer 1 electrical keepalives. (Ethernet calls this a link beat.) However, sometimes a cable is intact enough to maintain keepalives but not to pass data in both directions. This is a Unidirectional Link. Operating at Layer 2, Unidirectional Link Detection (UDLD) detects a unidirectional link by sending periodic hellos out to the interface. It also uses probes, which must be acknowledged by the device on the other end of the link.

UDLD has two modes: normal and aggressive. In normal mode, the link status is changed to Undetermined State if the hellos are not returned. In aggressive mode, the port is error-disabled if a unidirectional link is found. Aggressive mode is the recommended way to configure UDLD.

To enable UDLD on all fiber-optic interfaces, use the following command:

```
(config)# udld [enable ¦ aggressive]
```

Although this command is given at global config mode, it applies only to fiber ports.

To enable UDLD on nonfiber ports, give the same command at interface config mode.

To control UDLD on a specific fiber port, use the following command:

```
(config-if)# udld port {aggressive ¦ disable}
```

To reenable all interfaces shut by UDLD, use the following:

```
# udld reset
```

To verify UDLD status, use the following:

```
# show udld interface
```

Loop Guard

Loop Guard prevents loops that might develop if a port that should be blocking inadvertently transitions to the forwarding state. This can happen if the port stops receiving BPDUs (perhaps because of a unidirectional link or a software/configuration problem in its neighbor switch). When one of the ports in a physically redundant topology stops receiving BPDUs, the STP conceives the topology as loop-free. Eventually, the blocking port becomes designated and moves to forwarding state, thus creating a loop. With Loop Guard enabled, an additional check is made.

If no BPDUs are received on a blocked port for a specific length of time, Loop Guard puts that port into "loop inconsistent" blocking state, rather than transitioning to forwarding state. Loop Guard should be enabled on all switch ports that have a chance of becoming root or designated ports. It is most effective when enabled in the entire switched network in conjunction with UDLD.

To enable Loop Guard for all point-to-point links on the switch, use the following command:

```
(config)# spanning-tree loopguard default
```

To enable Loop Guard on a specific interface, use the following:

```
(config-if)# spanning-tree guard loop
```

Loop Guard automatically reenables the port if it starts receiving BPDUs again.

Troubleshooting STP

Some common things to look for when troubleshooting Spanning Tree Protocol include:

- **Duplex mismatch:** When one side of a link is half-duplex and the other is full-duplex. This causes late collisions and FCS errors.

- **Unidirectional link failure:** The link is up but data flows only in one direction. It can cause loops.

- **Frame corruption:** Physical errors on the line cause BPDUs to be lost, and the port incorrectly begins forwarding. This is caused by duplex mismatch, bad cable, or cable too long.

- **Resource errors:** STP is implemented in software, so a switch with an overloaded CPU or memory might neglect some STP duties.

- **Port Fast configuration errors:** Connecting a switch to two ports that have Port Fast enabled. This can cause a loop.

- **STP tuning errors:** Max age or forward delay set too short can cause a loop. A network diameter that is set too low causes BPDUs to be discarded and affects STP convergence.

Identifying a Bridging Loop

Suspect a loop if you see the following:

■ You capture traffic on a link and see the same frames multiple times.

■ All users in a bridging domain have connectivity problems at the same time.

■ There is abnormally high port utilization.

To remedy a loop quickly, shut redundant ports and then enable them one at a time. Some switches enable debugging of STP to help in diagnosing problems. The following commands are useful for isolating a bridging loop:

show interfaces

show spanning tree

show bridge

show process cpu

debug spanning tree

show mac address-table aging-time *vlan#*

show spanning-tree vlan *vlan#* **detail**

Spanning-Tree Best Practices

To optimize data flow in the network, design and configure Spanning Tree in the following ways:

■ Statically configure switches to be the primary and secondary root bridges by setting priority values.

■ Consider which interfaces will become designated and root ports (possibly set port priorities/path cost).

■ Tune STP using the tools detailed in this section.

■ Enable UDLD aggressive mode on all fiber interfaces.

■ Design STP domains that are as simple and contained as possible by using multilayer switches and routed links.

■ Use PVRST+ or MST for the fastest convergence times.

Confused by all the acronyms and STP features? Figure 3-3 shows the STP tools you might use in your network and where you might use them.

FIGURE 3-3
Example Switched
Topology
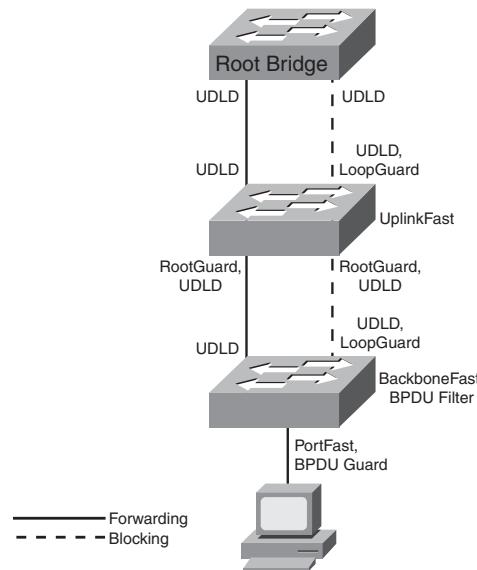
CCNP SWITCH Quick Reference

Denise Donohue

Warning and Disclaimer

This ebook is designed to provide information about networking. Every effort has been made to make this ebook 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 ebook.

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

Trademark Acknowledgments

All terms mentioned in this ebook 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 ebook 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 ebook, 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 ebook title and ISBN in your message.

We greatly appreciate your assistance.

Corporate and Government Sales

The publisher offers excellent discounts on this ebook 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 sales outside the United States please contact: **International Sales** international@pearsoned.com