CISCO SYSTEMS

Building Resilient IP Networks

The practical guide to building resilient and highly available IP networks

Kok-Keong Lee, CCIE® No. 8427
Fung Lim, CCIE No. 11970
Beng-Hui Ong

ciscopress.com

Building Resilient IP Networks

Kok-Keong Lee, CCIE No. 8427

Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Building Resilient IP Networks

Kok-Keong Lee

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

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

First Printing December 2005

Library of Congress Cataloging-in-Publication Number: 2004112420

ISBN: 1-58705-215-6

Warning and Disclaimer

This book is designed to provide information about building resilient IP networks. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The author, 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 book or from the use of the disks or programs that might accompany it.

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

Trademark Acknowledgments

All terms mentioned in this book 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 book should not be regarded as affecting the validity of any trademark or service mark.

Corporate and Government Sales

Cisco Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales.

For more information please contact:
U.S. Corporate and Government Sales 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside the U.S. please contact: **International Sales** international@pearsoned.com

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 from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through e-mail at feedback@ciscopress.com. Please make sure to include the book's title and ISBN in your message.

We greatly appreciate your assistance.

Publisher	John Wait
Editor-in-Chief	John Kane
Cisco Representative	Anthony Wolfenden
Cisco Press Program Manager	Jeff Brady
Production Manager	Patrick Kanouse
Development Editor	Dayna Isley
Editorial Assistant	Raina Han
Project Editor	Marc Fowler
Copy Editor	Keith Cline
Technical Editors	Roland Dobbins Ian Foo Mark Gallo Steve Moore
Book/Cover Designer	Louisa Adair
Composition	Mark Shirar
Indexer	WordWise Publishing

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: 31 0 20 357 1000
Fax: 31 0 20 357 1100

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
Capital Tower
168 Robinson Road
#22-01 to #29-01
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the **Cisco.com Web site at www.cisco.com/go/offices.**

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia • Czech Republic Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary • India • Indonesia • Ireland • Israel • Italy Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain • Sweden Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

Printed in the USA

About the Authors

Kok-Keong Lee, CCIE No. 8427, graduated from the University of Singapore, where he majored in computer and information systems sciences. From 1991 to 1993, Kok-Keong worked as a response center engineer for Hewlett-Packard, helping customers solve networking problems. From 1993 to 1999, Kok-Keong worked for IBM as a network systems engineer. Currently, he works as a consulting systems engineer for Cisco Systems, focusing on network architecture and IP core technologies. Kok-Keong lives with his wife in Singapore.

Fung Lim, CCIE No. 11970, joined Cisco Systems in October 1999 and has been working closely with Internet service providers (ISPs) in areas pertaining to network design and operations, as well as service provider security. He also assisted in the rollout of one of the first MPLS networks in Asia Pacific. He graduated from the National University of Singapore, majoring in electrical engineering with a minor in management of information systems. Fung Lim is also a Certified Information Systems Security Professional (CISSP No. 64090). Prior to joining Cisco Systems, Lim Fung was working as a systems engineer for Pacific Internet. Lim Fung lives with his wife in Singapore.

Beng-Hui Ong started his career as a system administrator for an ISP in Singapore. He then went to Digital Equipment Corporation (DEC) working with various ISPs to implement Internet application services. While at DEC, he implemented a large-scale video server with an ATM network for a local ADSL provider. Later, Beng-Hui worked for Cisco Systems as a consulting system engineer on network design and implemented various networks for ISPs and cable operators. Beng-Hui now holds a managerial position at Starhub Cable Vision, heading product management on broadband access products, value-added services, and new technologies development. Beng-Hui currently works for Cisco System as a product manager for the Broadband Edge And Midrange Routing Business Unit. Beng-Hui is married with a daughter and lives in Singapore.

About the Technical Reviewers

Mark Gallo is a systems engineering manager at Cisco Systems within the Channels organization. He has led several engineering groups responsible for positioning and delivering Cisco end-to-end systems, and for designing and implementing enterprise LANs and international IP networks. He has a bachelor of science degree in electrical engineering from the University of Pittsburgh and holds Cisco CCNP and CCDP certifications. Mark resides in northern Virginia with his wife, Betsy, and son, Paul.

Steve Moore, CCIE No. 4927, is an engineer with the Cisco IP Routing Deployment and Scalability Team, which is a part of the IOS Technologies Division of Cisco Engineering. He is responsible for discovering, testing, validating, and assisting in the customer deployment of new ideas relating to the scalability of routing protocols. He works closely with development, support, testing, and consulting groups within Cisco Systems as well as customers. A part of Steve's job is to educate, and he does so by working with customers directly and by writing white papers and speaking at various networker conferences. During the nine years Steve has worked at Cisco Systems, he has become known for his experience with routing protocols, as well as WAN technologies and optical networking.

Dedications

To my wife, Serene, for her endless support and unconditional sacrifice.
—KK

To my parents, and my wife, Andrea, for her constant support and encouragement.
—LF

To my wife, Nancy, for her constant support and to my daughter, Chloe, for her endless laughter and joy.
—BH

Acknowledgments

This has been an ambitious project, and we want to thank everyone who has helped us in this long process: our editor-in-chief, John Kane, for showing faith in unknown authors halfway around the globe, and for his patience and understanding; our development editor, Dayna Isley, for her patience and encouragement; the rest of the editorial team from Cisco Press for their assistance; and especially our reviewers, Roland Dobbins, Mark Gallo, and Steven Moore, for their precious time and comments. This book would not have been possible without your assistance.

KK would like to thank Paul Chen for his support to embark on this project. Finally, KK would like to thank Lim Fung and Beng-Hui for agreeing to collaborate on this project on such short notice.

Lim Fung would like to thank Sew Hoon for her guidance, encouragement, and support at Cisco Systems, and would like to thank KK and Beng-Hui for being such great working partners in the project.

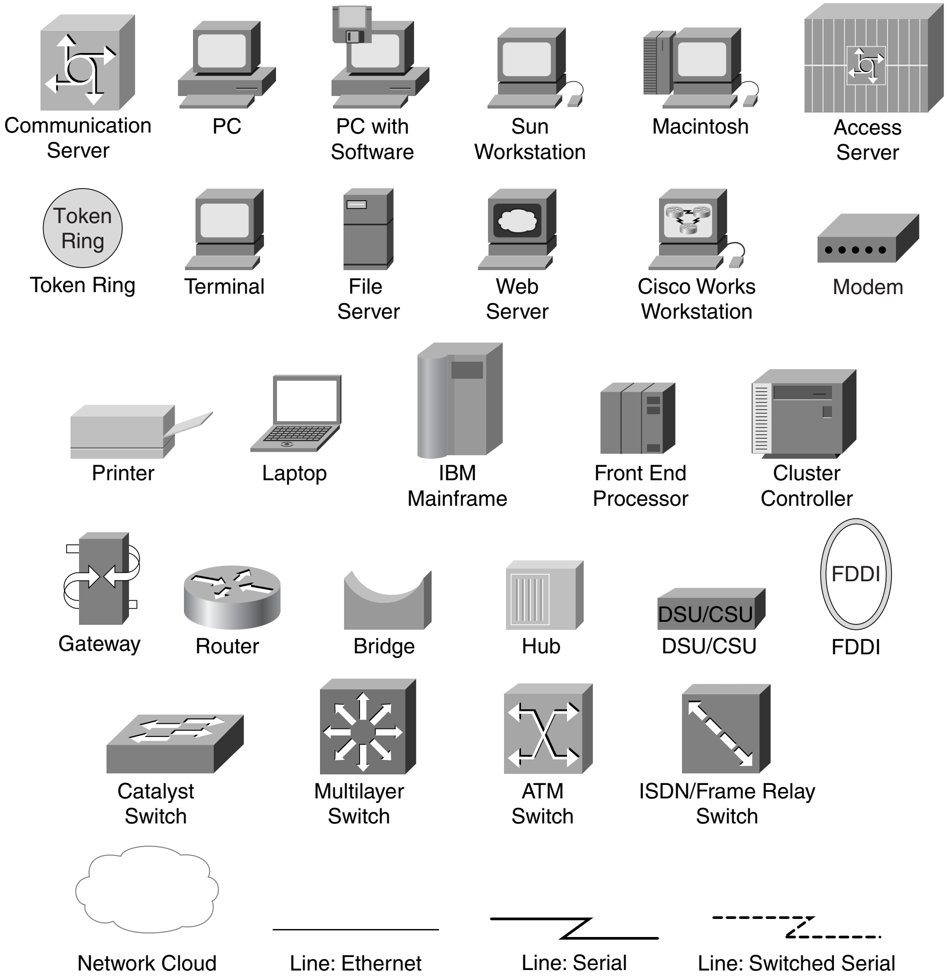
Beng-Hui would like to thank KK and Lim Fung for the invitation and for being great working partners. Beng-Hui would like to thank Paul Chen for his encouragement to start this project. Beng-Hui would also like to thank Sew Hoon for her constant encouragement and support at Cisco Systems.

Finally, the author team would like to thank Seo-Boon Ng for his ideas and contribution to the structure of this book. More importantly, we would like to thank him for his friendship.

Contents at a Glance

Contents

Icons Used in This Book



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Boldface** indicates commands and keywords that are entered literally as shown.
- *Italics* indicate arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets [ ] indicate optional elements.
- Braces { } indicate a required choice.
- Braces within brackets [{ }] indicate a required choice within an optional element.

Introduction

Over the years, many excellent books have been written on IP technology, and these books tend to focus on a specific topic, such as Open Shortest Path First (OSPF) or Border Gateway Protocol (BGP). However, building a resilient IP network requires a holistic approach to design and implementation. Various building blocks need to work together, most of which have seen enhancements and new features introduced. These features are scattered, mostly in the form of numerous Requests For Comments (RFCs), IOS features and documentations, and many Cisco Technical Assistance Center (TAC)-authored white papers.

The goal of this book is to provide a central resource for the features that focus on IP network resiliency. It covers a wide range of topics and serves as a foundation for anyone who needs a deeper understanding about IP network resiliency.

Who Should Read This Book?

This book is intended to increase your knowledge with respect to improving the resiliency of an IP network. It is not meant to be an introduction to IP network design. This book helps you understand what network availability means and introduces new features and tools that have been developed to improve resiliency in IP networks. Anyone who is involved in the design and implementation of a large-scale IP network and would like to understand ways to improve its resiliency will benefit from this book.

How This Book Is Organized

Although this book could be read cover to cover, it is designed to be flexible. You can easily move between chapters and sections of chapters to cover just the material that you need more information on.

- **Chapter 1, "Rise of the IP Transport System,"** provides a backdrop to the usage and new expectations of the IP network.
- **Chapter 2, "Establishing a High-Availability Network,"** takes a closer look at the real meaning of network resiliency and suggests a practical approach to tackling the "five-nines" availability challenge.
- **Chapter 3, "Fundamentals of IP Resilient Networks,"** serves as a refresher to the TCP/IP protocols and the various hardware and switching implementations that affect network resiliency. It also proposes a modular approach to solving the network resiliency problem.
- **Chapter 4, "Quality of Service,"** examines the relevancy of quality of service (QoS) with respect to network resiliency. Different QoS models and queuing techniques are examined, and a practical approach to a QoS architecture is presented.
- **Chapter 5, "Core Module"** examines the enhancements made to the various Interior Gateway Protocols to improve the resiliency of a network backbone.
- **Chapter 6, "Access Module,"** showcases the design guide for a resilient campus access network. Enhancements to the relevant protocol and first-hop gateways are highlighted.
- **Chapter 7, "Internet Module,"** focuses on resilient connectivity to the Internet. Improvements to the Exterior Gateway Protocol and addressing and routing solutions for redundancy are highlighted.
- **Chapter 8, "WAN Module,"** showcases improvements made to the various traditional WAN connectivity technologies.
- **Chapter 9, "Data Center Module,"** examines factors that influence data center network design. It highlights new technologies and a design guide to constructing the data center to support new application requirements.
- **Chapter 10, "Beyond Implementation: Network Management,"** looks at what more needs to be done beyond the basic configuration of hardware and routing protocols in a resilient IP network. The concept of proactive management is proposed, and the various supporting features are introduced.
- **Appendix A, "Calculating Network Availability,"** discusses mathematical calculations to evaluate network availability.
- **Appendix B, "RFCs Relevant to Building a Resilient IP Network,"** lists RFCs that are relevant to building resilient IP networks.
- **Appendix C, "The Cisco Powered Network Checklist,"** is an extract from the Cisco Powered Network data center best practices checklist.

This chapter covers the following topics:

- The Internet Explosion
- Next-Generation IP Applications
- MPLS: New Kid on the Block
- Next-Generation IP Transport System
- Continuous Improvements of Protocols

This chapter covers the following topics:

- Revisiting IP, TCP, and UDP
- Device-level resiliency
- Impact of different switching paths
- Key principles for designing resilient networks

CHAPTER 3

Fundamentals of IP Resilient Networks

Building a resilient IP network requires more than just putting in duplicate devices and turning on every feature available. A network manager must understand how those relevant protocols work, the architecture of the hardware being deployed, how to predict traffic growth, and how to evolve the network architecture.

This chapter revisits the mechanics of three major protocols: IP, TCP, and UDP. Especially of interest to you will be some of the behaviors of these protocols with respect to network resiliency. This chapter also examines how improvements in hardware features help contribute to network resiliency. Most important, you must have a strategy in place, with a few important design principles that we have learned from those who have done it.

Revisiting IP, TCP, and UDP

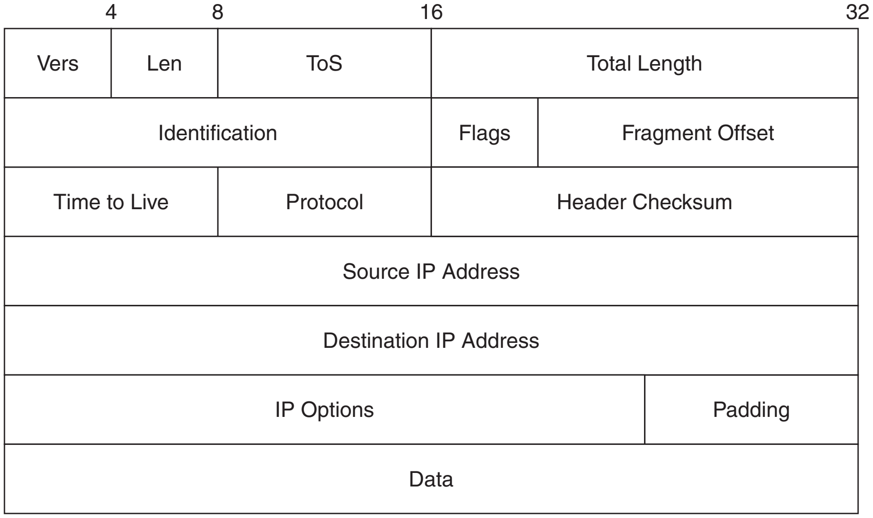
This section examines the IP, TCP, and UDP protocols. The building of a resilient network is all about managing these protocols. So a good understanding of how they function is crucial to understanding and resolving the basic challenges that you face.

Internet Protocol

The IP protocol corresponds to the network layer (Layer 3) of the Open Systems Interconnection (OSI) reference model. Its function is to transmit a data block, called a *datagram*, from a source to a destination. Part of its functionality includes fragmentation and reassembly. As a protocol, IP does not provide for data integrity, reliability of transmission, or sequencing.

Figure 3-1 illustrates the IP datagram format.

Figure 3-1 *The IP Datagram Format*



The nature of how IP behaves and the functions of some of the fields illustrated directly impact the resiliency of the network, as follows:

- IP cannot handle signal degradation during transmission. For example, when the quality of a pair of fiber has degraded and is causing a high burst error rate, IP continues to transmit packets, even though it is sending garbage. Therefore, selection of Layer 2 technology is important because the right one can complement this deficiency. For example, a Packet over SONET (PoS) interface is always preferred as compared to Gigabit Ethernet. A PoS interface monitors the quality of its transmission and can declare itself unusable so that a reroute can take place.
- IP does not retransmit data in the event of a missing packet. It treats every packet as an independent entity and has no knowledge of whether other related packets are transmitted successfully. It is the job of the upper-layer protocol to deal with this. Therefore, selecting the right upper-layer protocol is crucial for applications that want resilient capability.
- Because IP treats every packet independently, packets for an application might not arrive at the destination in sequence. This happens when multiple paths are made available to a router along the transmission. Although it is the duty of the destination host to handle this situation, this might not be ideal for applications such as voice. Therefore, the forwarding decisions the router makes, whether per-destination load balancing or per-packet load balancing, directly impact applications.

- IP can fragment and reassemble packets. And it can work with different transmission technologies, such as Ethernet, PoS, and so on. These technologies have different maximum transmission unit (MTU) sizes, and traversing from one to another might require the IP protocol to break bigger packets into numerous smaller ones. However, if in the case of a misconfiguration, fragmentation taxes the processing power of the router, this might impact the performance of the network. When critical resources such as CPU of the router is taxed, resiliency is compromised. Therefore, fragmentation is one area that you want to avoid.
- The Type of Service (ToS) field has mostly been ignored until recently when the concept of quality of service (QoS) has become critical. QoS is the basis of identifying different classes of traffic within a network. In the event of network congestion, certain classes of traffic might be deemed less important and be dropped by the network. In this case, network service is affected for some users. Chapter 4, "Quality of Service," discusses QoS in more detail.
- The Options field is also one that is mostly ignored but has an impact on the performance of the network, which ultimately affects network services. Because of the specification of the IP protocol, any IP packet that has an Options field filled must be examined for further action. The packet has to be sent to the router's CPU for processing before being forwarded. In this case, the CPU is being hit by extra work and might potentially slow down the router.
- The length of the IP header is fixed, but the data portion is variable. This means that its efficiency might not be great when the actual payload is small. What this means is that the size of the payload traversing the network does impact the efficiency of the network. Although a larger payload is always preferred in applications such as a file transfer, some applications (Voice over IP [VoIP], for example) send traffic in small packet sizes. Therefore, it is important to have a traffic matrix utilizing the network so that a more accurate capacity can be planned for.

As you might have realized by now, one just needs to review the protocol more thoroughly to realize its impact on network resiliency. This is especially so when some of the fields are used to implement features such as QoS. If you are thinking that the IP protocol is dated and no more work is being done on it, you just might be proven wrong. One just needs to trace the development work from the various RFCs to know that experts have worked on it continuously. To stay informed of the latest development, visit the Internet Engineering Task Force (IETF) website at http://www.ietf.org.

Transmission Control Protocol

The TCP protocol provides reliable transmission of data over an IP network. It works around transmission errors by resending packets. TCP corresponds to the transport layer (Layer 4) of the OSI reference model. As a complement to the IP protocol, it provides functionality that is missing in IP.

Figure 3-2 shows the TCP packet format.

Figure 3-2 *TCP Packet Format*

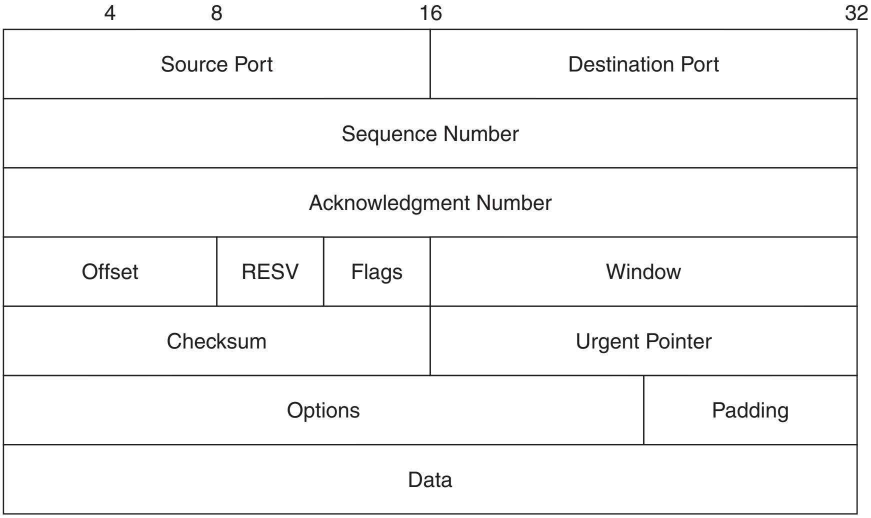


Figure 3-2 illustrates the TCP packet format. Similar to IP, the nature of how TCP behaves and the functions of the fields within the TCP packet have the following implications with respect to resiliency:

- TCP delivers an unstructured stream of bytes identified by sequence numbers, which is called *stream data transfer*. This service benefits applications because they do not have to chop data into blocks before handing it off to TCP. TCP accomplishes this by grouping bytes into segments and passing them to IP for delivery.
- TCP provides connection-oriented, end-to-end reliable packet delivery through an IP network. It does so by sequencing bytes with a forwarding acknowledgment number indicating to the destination the next byte the source expects to receive. Bytes not acknowledged within a specified time period are retransmitted. The reliability mechanism of TCP allows hosts to deal with lost, delayed, duplicate, or out-of-order packets that might happen as they move through an IP network. This ability to recover from errors is sometimes exploited by network managers when dealing with network congestion via features such as weighted random early detection (WRED), as discussed in Chapter 4.
- When sending acknowledgments back to the source, the receiving TCP process indicates the highest sequence number it can receive without overflowing its internal buffers. This flow-control capability of TCP dictates bandwidth consumption, and in fact, affects how fast an application appears to a user.

- TCP processes can both send and receive at the same time. This means fewer flows to keep track of within the network, which is good in the event that you need to deal with per-flow types of operations (for example, accounting). With fewer flows to keep track of, the network saves on critical resources, such as memory, that are important in maintaining the resiliency.
- The TCP process also allows for multiplexing, which enables numerous upper-layer conversations to take place over a single connection. Again, this decreases the number of flows within the network.

Beside these points, two other important characteristics of TCP need special attention: the three-way handshake and the sliding window.

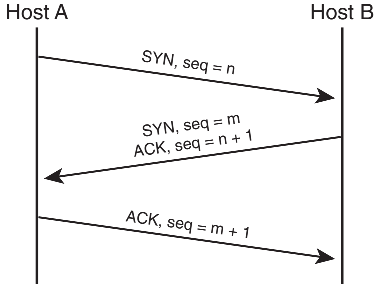
TCP Three-Way Handshake

One of the important characteristics of TCP is the way it establishes a connection between two hosts. The so-called *three-way handshake* is a signature of TCP and is a basis of how some technology works. One good example is a server load-balancing technique, which Chapter 9, "Data Center Module," discusses.

Because of the way resources have to be allocated for a three-way handshake to work, many network attacks exploit this characteristic to compromise the resiliency of the network.

When a pair of devices wants to initiate a TCP connection, they undergo a process known as a three-way handshake. The initiating side sends a SYN packet to the peer, which replies with an ACK and a SYN of its own. Sequence number negotiation occurs here. Finally, the initiator sends an ACK for the peer's SYN. Figure 3-3 illustrates the TCP three-way handshake.

Figure 3-3 *TCP Three-Way Handshake*



After the connection has been established, data begin to flow between the two hosts. The two hosts then maintain states of the connection, which consume memory resources. Therefore, in a form of a denial-of-service (DoS) attack, the attacker issues numerous SYN requests to a target host, which cause a SYN_ACK response from the target. The target host will soon run out of resources trying to maintain the various SYN requests and eventually not be able to process legitimate SYN requests from other hosts. Even worse, the target host might not even function properly at all because of a lack of memory resources. Therefore, it is critical that important resources within the network, such as the servers and routers, be protected from attacks that exploit the three-way handshake.

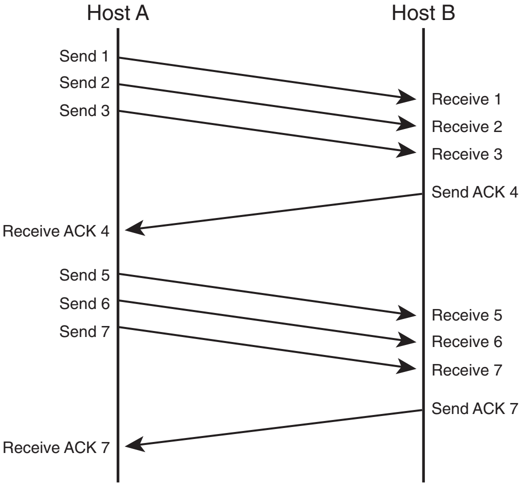
TCP Sliding Window

The TCP sliding window is another important characteristic of the TCP protocol. It is a flow-control mechanism used by TCP to regulate the amount of traffic received from a sender at any one time. Data received from the sender is stored in a buffer that feeds the data to an application. If the application is slow in removing the data from the buffer, TCP slows down the sender in transmission. On the other hand, if the application can retrieve data faster than the buffer can be filled up, TCP informs the sender to speed up. The speed to send is determined through a negotiated entity called a *window size*. Window size refers to the number of messages that can be transmitted while awaiting an acknowledgment. After a host transmits the window-size number of messages, it must receive an acknowledgment before any more messages can be sent.

With a window size of 1, each segment must be acknowledged before another segment is transmitted. This methodology results in inefficient use of bandwidth by the hosts.

Figure 3-4 shows a window size of 3.

Figure 3-4 *TCP Sliding Window*



A larger window size allows more data to be transmitted pending acknowledgment, which improves the performance of data transmission, as shown in Figure 3-4.

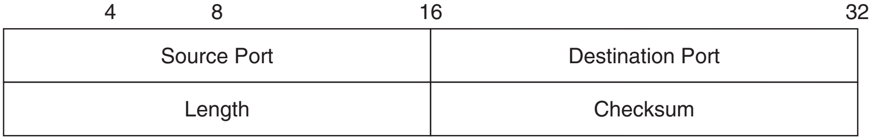
TCP uses expectation acknowledgments, meaning that the acknowledgment number refers to the packet that is expected next. The "sliding" part of "sliding window" refers to the fact that the window size is negotiated dynamically during the TCP session. Because it is dynamic, it can be increased or reduced during the life of the conversation. Because of this characteristic, the sliding window may be exploited to slow down an application when network congestion occurs. This allows for other more important applications to get service from the network and, hence, improves resiliency for those classes of traffic.

User Datagram Protocol

As opposed to the TCP, User Datagram Protocol (UDP) is a connectionless protocol. It is simpler than TCP in implementation and does not have overheads such as the three-way handshake. UDP corresponds to the transport layer (Layer 4) of the OSI reference model. As a complement to the IP protocol, it serves as a transport mechanism to applications that need efficiency in transmission rather than reliability, and is used when error correction or even resend does not make sense. Examples of such applications are video streaming and voice applications.

Figure 3-5 shows the UDP datagram format.

Figure 3-5 *UDP Datagram Format*



The nature of how the UDP protocol behaves and its format have the following implications with respect to resiliency:

- As you can see, the UDP datagram format is much simpler than that of TCP. It does not have most of the characteristics of TCP recovery. Because UDP does not provide a recovery function, it depends on higher-layer applications to do so.
- You might wonder why you should implement UDP if it does not provide for a reliable transmission. UDP does one thing better than TCP: performance. It is useful in situations where performance is preferred over reliability of the protocol. Some applications work on the basis of short message passing with a small chance of error. The application might provide its own error-recovery mechanism. The following applications use UDP:
  - — Network File System (NFS)
  - — Simple Network Management Protocol (SNMP)
  - — Domain Name System (DNS)
  - — Trivial File Transfer Protocol (TFTP)

- UDP does not have a flow control mechanism like that of TCP. In other words, features such as WRED will not work on the UDP protocol. Therefore, in the event of network congestion, an application that uses UDP cannot be slowed down by the network. The throttling mechanism must come from the application itself. From a network-resiliency perspective, these applications become more difficult to control. Although you may choose to police the transmission rate of these applications, dropping too many packets might affect its quality.
- If you are not familiar with UDP, familiarize yourself now. Most video and VoIP applications run on UDP. Learning to deal with UDP will prove important as you try to keep your voice network running continuously.

So far, this chapter has highlighted the characteristics of the major protocols such as IP, TCP, and UDP and how their behaviors impact the resiliency of the network. Another area that contributes to the resiliency of the network is the high-availability feature of the hardware that supports the running of these protocols.

This chapter now looks at how fast the hardware can recover from a failure so as not to affect the running of these protocols, which ultimately affect the applications that are running on them. The following section discusses the development of these hardware features and how the improvements seek to complement the behaviors of the major protocols in supporting a resilient IP network.

Device-Level Resiliency

Device reliability directly impacts the overall resiliency of the network. Many protocol enhancements have been based on improvements made to device-level resiliency. The sections that follow trace the development work done on device-level resiliency, including the following:

- Online insertion and removal (OIR)
- Single line card reload
- High system availability (HSA)
- Route processor redundancy (RPR)
- Route processor redundancy plus (RPR+)
- Stateful switchover (SSO)
- Nonstop forwarding (NSF)

Online Insertion and Removal (OIR)

The online insertion and removal (OIR) feature was developed to enable network managers to replace faulty parts on a hardware device without affecting system operation. For example, when a replacement card is inserted, power is available on the card, and it initializes itself to start working. This feature is also referred to as *hot swap capability.*

Hot swap capability on a card, for example, allows the system to determine when a change occurs in the unit's physical configuration, and reallocates the unit's resources to allow all interfaces to function adequately. This feature allows interfaces on the card to be reconfigured while other interfaces on the router remain unchanged.

The software performs the necessary tasks involved in handling the removal and insertion of the card. A hardware interrupt is sent to the software subsystem when a hardware change is detected, and the software reconfigures the system accordingly. When a card is inserted, it is analyzed and initialized in such a way that the end user can configure it properly. The initialization routines used during OIR are the same as those called when the router is powered on. System resources are allocated to the new interface so that it can begin functioning. Likewise, when a card is removed, the resources associated with the empty slot must either be freed or altered to indicate the change in its status.

Although the concept of OIR might sound simple enough, it is important to know what exactly is happening within the hardware. When an OIR is performed on a Cisco router, the following will happen:

1. The router rapidly scans the backplane for configuration changes.
2. The router initializes all newly inserted interfaces and places them in the administratively shutdown state. This is the default.
3. The router brings all previously configured interfaces on the card back to the state they were in when they were removed. Any newly inserted interfaces are put in the administratively shutdown state.

The effect on the routing table of the router is that routes learned through the removed interface are deleted. Likewise, the Address Resolution Protocol (ARP) cache is selectively flushed. If the card is reinserted into the same slot from which it was removed, or if an identical card is inserted in its place, many of the control blocks from the previous installation are reused.

Almost all Cisco routers today support OIR, but some hardware requires special attention. One good example is the FlexWAN module on the Cisco 7600 series router. The FlexWAN module acts as a carrier module for you to insert port adaptors such as those used on the 7200 and 7500 series into the 7600 series router. Before a port adaptor is removed or inserted into the FlexWAN module, the module has to be removed. In this case, the FlexWAN module is OIR capable, but not the port adaptors. On the other hand, this caveat does not apply to the Port Adaptor Carrier module on the Cisco 7300 series.

If a router does not support OIR and an interface is inserted or removed from it, it might impact traffic being processed by the router, and, in a worse situation, the router might simply reboot. Therefore, it is crucial that you read the installation guide of all modules and port adaptors carefully. A mistake can be costly; it might mean permanent damage done to the card, or worse, the router.

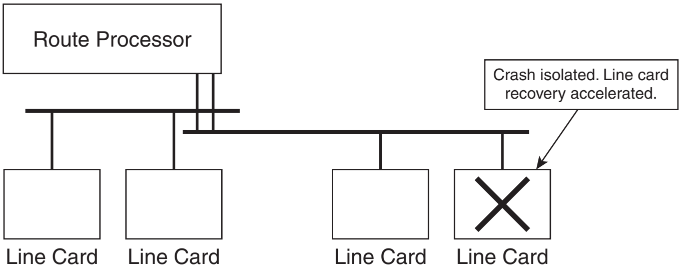
Single Line Card Reload

Before the introduction of the Cisco 7500 single line card reload feature, the only way to correct a line card hardware failure or a severe software error for one line card on a Cisco 7500 series router was to execute a Cbus Complex—a process that reloaded every line card on the network backplane. The time taken to complete the Cbus Complex was often inconvenient, and no network traffic could be routed or switched during the Cbus Complex process.

The Cisco 7500 single line card reload feature enables users to correct a line card failure on a Cisco 7500 series router by automatically reloading the microcode on a failed line card. During the single line card reload process, all physical lines and routing protocols on the other line cards of the network backplane remain active. A single line card reload is also significantly faster than the Cbus Complex process.

Figure 3-6 illustrates the concept of the single line card reload.

Figure 3-6 *Single Line Card Reload*



Because the Cisco 7500 single line card reload feature reloads only the line card with the hardware failure rather than all the line cards on the network backplane, the active line cards can continue to forward network traffic.

Use the **show running-config** command to verify that single line card reloading has been successfully enabled on the Cisco 7500 series router. If the "service single-slot-reload-enable" line appears in the command output, the Cisco 7500 single line card reload feature is enabled. If this line does not appear in the command output, the Cisco 7500 single line card reload feature is disabled.

Example 3-1 demonstrates how to configure single line card reload.

Example 3-1 *Configuring Single Line Card Reload*

```
7507#config t
Enter configuration commands, one per line.  End with CNTL/Z.
7507(config)#service single-slot-reload-enable
```

The output in Example 3-2 shows the status of the configuration.

Example 3-2 *Verifying Single Line Card Reload Configuration Status*

```
7507#show running-config
8w1d: %SYS-5-CONFIG_I: Configured from console by consoleun
Building configuration...
Current configuration : 3587 bytes
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service multiple-config-sessions
service single-slot-reload-enable
```

For the rest of the chassis-based routers that were developed after the Cisco 7500 series (for example, the Cisco 12000 and the 7600 series routers), this capability is a default rather than a feature that needs configuration. You may reload a line card on the Cisco 12000, 10000, and the 7600 series router without affecting the functioning of the router. It is also available on the Catalyst range of switches.

High System Availability

The HSA feature first appeared on the Cisco 7507 and Cisco 7513 routers. This feature enables you to install two route processor (RP) cards in a single router to improve system availability. HSA is not available on routers such as the Cisco 7200 series because it supports only one processor.

The two RP cards in a router provide the most basic level of increased system availability through the so-called *cold restart feature*. A cold restart means that when one RP card fails, the other RP card reboots the router. In this way, your router is never in a failed state for very long, and, hence, system availability is increased.

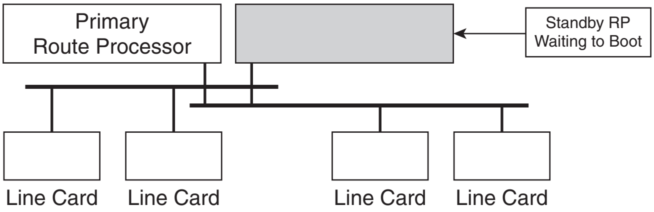
When one RP card takes over operation from another in HSA, system operation is interrupted. This change is similar to issuing the **reload** command. The following events occur when one RP card fails and the other takes over:

1. The router stops passing traffic.
2. Route information is lost.
3. All connections are lost.

The backup or "slave" RP card becomes the active or "master" RP card that reboots and runs the router. Thus, the slave has its own image and configuration file so that it can act as an independent processor.

Figure 3-7 illustrates the concept of HSA.

Figure 3-7 *High System Availability*



In the HSA process, the time from initial failure to first packet transmission is the aggregate of the times required for each of the following:

1. Identify failure.
2. Load and boot software on standby RP.
3. Load new configuration on standby RP.
4. Reset and reload line cards.
5. Load new configuration on line cards.
6. Learn routes, pass keepalive message, and forward traffic.
7. Route convergence at routing protocol level.

The HSA way of resiliency is also called *cold standby*, which implies that the entire system will lose function for the duration of the restoration via cold restart. All traffic flowing through the router is lost during this time. The benefit of using cold standby is that the device will restart without manual intervention by rebooting with the standby RP taking control of the router.

Taking into consideration how HSA operates, there are two common ways to use HSA, as follows:

- **Simple hardware backup**—Use this method to protect against an RP card failure. With this method, you configure both RP cards with the same software image and configuration information. Also, you configure the router to automatically synchronize configuration information on both cards when changes occur.
- **Software error protection**—Use this method to protect against software errors in a particular release. With this method, you configure the RP cards with different software images but with the same configuration information. If you are using new or experimental software, you might want to consider using the software error-protection method.

You can also use HSA for advanced implementations. For example, you can configure the RP cards with the following:

- Similar software versions, but different configuration files
- Different software images and different configuration files
- Widely varied configuration files (For example, various features or interfaces can be turned off and on per card.)

Example 3-3 shows the configuration steps for HSA setup.

Example 3-3 *HSA Setup Configuration*

```
7507#config t
Enter configuration commands, one per line.  End with CNTL/Z.
7507(config)#redundancy
7507(config-red)#mode hsa
```

When the system has been successfully configured, you can verify the HSA status using the **show redundancy** command, as demonstrated in Example 3-4.

Example 3-4 *Verifying HSA Status*

```
7507#show redundancy
Redundant System Information :
------------------------------
       Available system uptime = 8 weeks, 1 day, 10 hours, 11 minutes
Switchovers system experienced = 0
              Standby failures = 1
        Last switchover reason = none
                 Hardware Mode = Duplex
    Configured Redundancy Mode = hsa
     Operating Redundancy Mode = hsa
              Maintenance Mode = Disabled
                Communications = Down      Reason: Simplex mode
Current Processor Information :
-------------------------------
               Active Location = slot 2
        Current Software state = ACTIVE
       Uptime in current state = 8 weeks, 1 day, 10 hours, 10 minutes
                 Image Version = Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-PV-M), Version 12.0(27)S1, EARLY DEPLOYMENT RELEASE )
Technical Support: http://www.cisco.com/techsupport
Copyright©  1986-2004 by cisco Systems, Inc.
Compiled Mon 23-Feb-04 01:14 by nmasa
                          BOOT = ,1;disk0:rsp-pv-mz.120-21.S8,1;
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2102
Peer (slot: 3) information is not available because it is in 'DISABLED' state
7507#
```

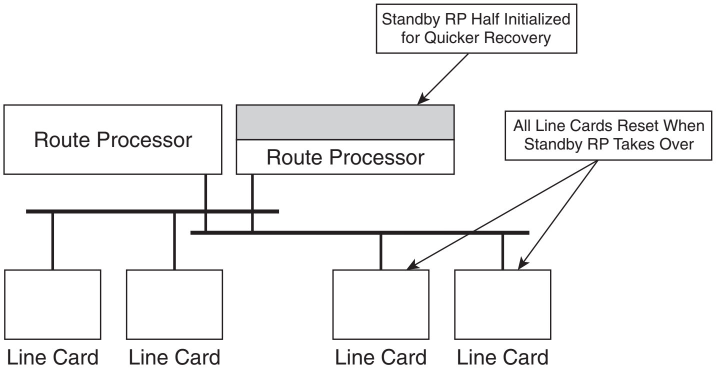
Route Processor Redundancy

The route processor redundancy (RPR) feature provides an improvement to the HSA feature. HSA enables a system to reset and use a standby RP in the event of a failure of the active RP.

With RPR, you can achieve a quicker switchover between an active and standby RP in the event of a fatal error on the active RP. When you configure RPR, the standby RP loads a Cisco IOS image upon boot up and initializes itself in standby mode. In the event of a fatal error on the active RP, the system switches to the standby RP, which reinitializes itself as the active RP, reloads all the line cards, and restarts the system.

Figure 3-8 illustrates the concept of RPR.

Figure 3-8 *Route Processor Redundancy (RPR)*


Standby RP Half Initialized for Quicker Recovery
Route Processor
Route Processor
All Line Cards Reset When Standby RP Takes Over
Line Card
Line Card
Line Card
Line Card

The RPR feature eliminates the following in the HSA switchover process, and, thus, reduces the failure recovery time:

- Step 1: Load and boot software on standby RP.
- Step 2: Load new configuration on standby RP.

The recovery time is now reduced because the standby RP has already started the boot-up process before taking control of the router. This is also called *warm standby mode*.

Example 3-5 demonstrates how to configure RPR.

Example 3-5 *Configuring RPR*

```
7507#config t
8w1d: %SYS-5-CONFIG_I: Configured from console by console t
Enter configuration commands, one per line.  End with CNTL/Z.
7507(config)#redundancy
7507(config-red)#mode rpr
```

Example 3-6 shows sample output of the RPR status.

Example 3-6 *Verifying RPR Configuration Status*

```
7507#show redundancy
Redundant System Information :
------------------------------
       Available system uptime = 8 weeks, 1 day, 10 hours, 25 minutes
Switchovers system experienced = 0
              Standby failures = 2
        Last switchover reason = none
                 Hardware Mode = Duplex
    Configured Redundancy Mode = rpr
     Operating Redundancy Mode = rpr
              Maintenance Mode = Disabled
                Communications = Down      Reason: Simplex mode
Current Processor Information :
-------------------------------
               Active Location = slot 2
        Current Software state = ACTIVE
       Uptime in current state = 8 weeks, 1 day, 10 hours, 24 minutes
                 Image Version = Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-PV-M), Version 12.0(27)S1, EARLY DEPLOYMENT RELEASE )
Technical Support: http://www.cisco.com/techsupport
Copyright©  1986-2004 by cisco Systems, Inc.
Compiled Mon 23-Feb-04 01:14 by nmasa
                          BOOT = ,1;disk0:rsp-pv-mz.120-27.S1.bin,1;
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2102
Peer (slot: 3) information is not available because it is in 'DISABLED' state
7507#
```

Route Processor Redundancy Plus

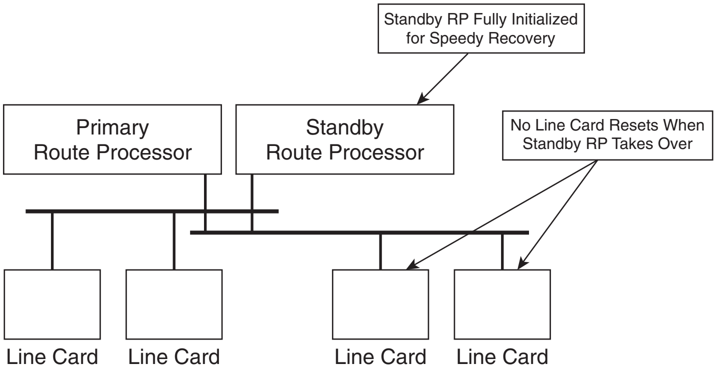
Building on the RPR feature, the RPR+ feature further eliminates steps in the HSA switchover process:

- Step 3: Reset and reload line cards.
- Step 4: Load new configuration on line cards.

RPR+ on the router keeps the line cards up and running during the switchover. The line cards will not be reloaded or reinitialized, and they continue to forward traffic. This feature reduces the route processor switchover time by 90 percent (down to 30 to 40 seconds) compared to RPR.

Figure 3-9 illustrates the concept of RPR+.

Figure 3-9 *Route Processor Redundancy Plus (RPR+)*


Standby RP Fully Initialized for Speedy Recovery
Primary Route Processor
Standby Route Processor
No Line Card Resets When Standby RP Takes Over
Line Card
Line Card
Line Card
Line Card

Example 3-7 demonstrates how to configure RPR+.

Example 3-7 *Configuring RPR+*

```
7507#config t
Enter configuration commands, one per line.  End with CNTL/Z.
7507(config)#hw-module slot 2 image disk0:rsp-pv-mz.120-27.S1.bin
7507(config)#hw-module slot 3 image disk0:rsp-pv-mz.120-27.S1.bin
7507(config)#redundancy
7507(config-red)#mode rpr-plus
```

The output in Example 3-8 shows the status of the router after RPR+ has been configured. Notice the peer status shows STANDBY COLD-CONFIG.

Example 3-8 *Verifying RPR+ Configuration Status*

```
7507#show redundancy
Redundant System Information :
------------------------------
       Available system uptime = 8 weeks, 1 day, 10 hours, 38 minutes
Switchovers system experienced = 0
              Standby failures = 4
        Last switchover reason = none
                 Hardware Mode = Duplex
    Configured Redundancy Mode = rpr-plus
     Operating Redundancy Mode = rpr-plus
              Maintenance Mode = Disabled
                Communications = Up
Current Processor Information :
-------------------------------
               Active Location = slot 2
        Current Software state = ACTIVE
       Uptime in current state = 8 weeks, 1 day, 10 hours, 37 minutes
                 Image Version = Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-PV-M), Version 12.0(27)S1, EARLY DEPLOYMENT RELEASE
```

Example 3-8 *Verifying RPR+ Configuration Status (Continued)*

```
SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright©  1986-2004 by cisco Systems, Inc.
Compiled Mon 23-Feb-04 01:14 by nmasa
                          BOOT = ,1;disk0:rsp-pv-mz.120-27.S1.bin,1;
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2102
Peer Processor Information :
----------------------------
              Standby Location = slot 3
        Current Software state = STANDBY COLD-CONFIG
       Uptime in current state = 1 minute
                 Image Version = Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-PV-M), Version 12.0(27)S1, EARLY DEPLOYMENT RELEASE
SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright©  1986-2004 by cisco Systems, Inc.
Compiled Mon 23-Feb-04 01:14 by nmasa
                          BOOT = ,1;disk0:rsp-pv-mz.120-27.S1.bin,1;
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2102

7507#
```

Stateful Switchover

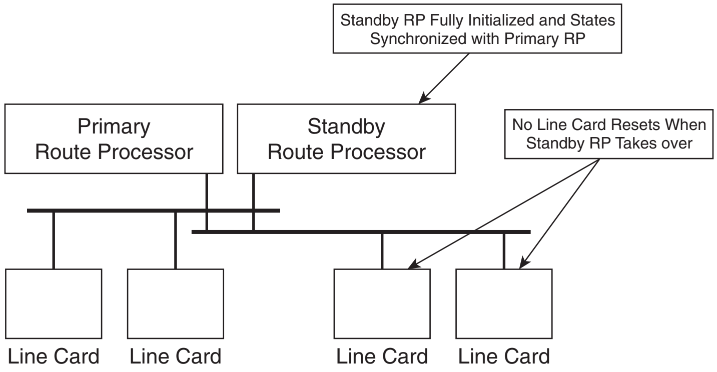
The stateful switchover (SSO) feature, which is based on RPR+, reduces the time in Step 5 in the HSA switching process:

- Step 5: Learn routes, pass keepalive message, and forward traffic.

The SSO allows the active RP to pass the necessary state information of key routing and interface protocols to the standby RP upon switchover, thereby reducing the time for the standby RP to learn and converge routes.

Figure 3-10 illustrates the concept of SSO.

Figure 3-10 *Stateful Switchover (SSO)*


Standby RP Fully Initialized and States Synchronized with Primary RP
Primary Route Processor
Standby Route Processor
No Line Card Resets When Standby RP Takes over
Line Card
Line Card
Line Card
Line Card

For any hardware device that is running SSO, both RPs must be running the same software and configuration so that the standby RP is always ready to assume control following a fault on the active RP. The configuration information is synchronized from the active RP to the standby RP at startup and whenever changes to the active RP configuration occur. Following an initial synchronization between the two processors, SSO maintains RP state information between them, including forwarding information.

During switchover, system control and routing protocol execution are transferred from the active processor to the standby processor. The time required by the device to switch over from the active to the standby processor ranges from platform to platform. The Cisco 12000 series switches immediately and can achieve zero packets dropped, whereas the Cisco 7500 series may take between 5 and 10 seconds.

Example 3-9 demonstrates the configuration required for SSO.

Example 3-9 *Configuring SSO*

```
7507#config t
Enter configuration commands, one per line.  End with CNTL/Z.
7507(config)#hw-module slot 2 image disk0:rsp-pv-mz.120-27.S1.bin
7507(config)#hw-module slot 3 image disk0:rsp-pv-mz.120-27.S1.bin
7507(config)#redundancy
7507(config-red)#mode sso
```

The output in Example 3-10 shows the status of the router after it has been configured for SSO. Notice the peer status shows STANDBY HOT.

Example 3-10 *Verifying SSO Configuration Status*

```
7507#show redundancy
Redundant System Information :
------------------------------
       Available system uptime = 8 weeks, 1 day, 10 hours, 43 minutes
Switchovers system experienced = 0
```

Example 3-10 *Verifying SSO Configuration Status (Continued)*

```
              Standby failures = 5
        Last switchover reason = none
                 Hardware Mode = Duplex
    Configured Redundancy Mode = sso
     Operating Redundancy Mode = sso
              Maintenance Mode = Disabled
                Communications = Up
Current Processor Information :
-------------------------------
               Active Location = slot 2
        Current Software state = ACTIVE
       Uptime in current state = 8 weeks, 1 day, 10 hours, 42 minutes
                 Image Version = Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-PV-M), Version 12.0(27)S1, EARLY DEPLOYMENT RELEASE
SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright©  1986-2004 by cisco Systems, Inc.
Compiled Mon 23-Feb-04 01:14 by nmasa
                          BOOT = ,1;disk0:rsp-pv-mz.120-27.S1.bin,1;
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2102
Peer Processor Information :
----------------------------
              Standby Location = slot 3
        Current Software state = STANDBY HOT
       Uptime in current state = 0 minutes
                 Image Version = Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-PV-M), Version 12.0(27)S1, EARLY DEPLOYMENT RELEASE
SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright©  1986-2004 by cisco Systems, Inc.
Compiled Mon 23-Feb-04 01:14 by nmasa
                          BOOT = ,1;disk0:rsp-pv-mz.120-27.S1.bin,1;
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2102

7507#
```

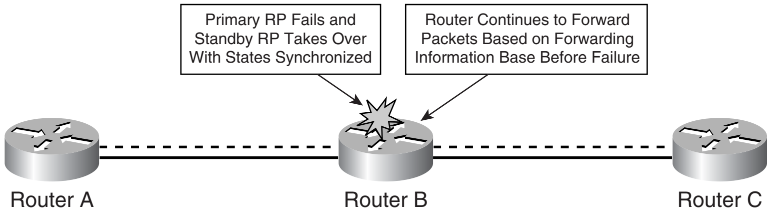
Nonstop Forwarding

The NSF feature works with the SSO feature in the Cisco IOS Software. Whereas SSO solves an internal problem (RP failure), NSF prevents some external event that might prove harmful to the network from occurring.

Usually when a networking device restarts, all routing peers of that device detect that the device went down and then came back up. This transition results in a *routing flap*, which could spread across multiple routing domains. Although the device might be forwarding traffic, routing flaps caused by the switchover create routing instabilities, which are detrimental to overall network performance. NSF helps to suppress routing flaps in SSO-enabled devices, thus reducing network instability.

Figure 3-11 illustrates the concept of NSF.

Figure 3-11 *Nonstop Forwarding (NSF)*


Primary RP Fails and Standby RP Takes Over With States Synchronized
Router Continues to Forward Packets Based on Forwarding Information Base Before Failure
Router A
Router B
Router C

NSF allows for the forwarding of data packets to continue along known routes while the routing protocol information is being restored following a switchover. With NSF, routing peers do not experience routing flaps. Data traffic is forwarded through the line cards while the standby RP assumes control from the failed active RP during a switchover. The capability of line cards to remain up through a switchover and to be kept current with the Forwarding Information Base (FIB) on the active RP is key to NSF operation.

For NSF to function, protocols and applications must be high-availability (HA) aware. A feature or protocol is HA aware if it maintains, either partially or completely, undisturbed operation through an RP switchover. For some HA-aware protocols and applications, state information is synchronized from the active to the standby processor.

For NSF to work, enhancements to the routing protocols such as Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS), and Border Gateway Protocol (BGP) have been made. These enhancements are also known as *graceful restart extensions* to the protocols. With the enhancements, these protocols have been made NSF capable and aware, which means that routers running these protocols can detect a switchover and take the necessary actions to continue forwarding network traffic and to recover route information from the peer devices. For example, the IS-IS protocol can be configured to use state information that has been synchronized between the active and the standby RP to recover route information following a switchover instead of information received from peer devices.

A device is said to be NSF capable if it has a redundant RP and has been configured to support NSF; therefore, it would rebuild routing information from NSF-aware or NSF-capable neighbors. A networking device is said to be NSF aware if it is running NSF-compatible software but it does not have redundant RP (for example, the Cisco 7200 series router).

For NSF to work, Cisco Express Forwarding (CEF) has also been enhanced, because each protocol depends on CEF to continue forwarding packets during switchover while the routing protocols rebuild the Routing Information Base (RIB) tables. When the routing protocols have converged, CEF updates the FIB table and removes stale route entries. CEF, in turn, updates the line cards with the new FIB information. CEF is discussed in detail in the section "Cisco Express Forwarding Switching" later in this chapter.

Example 3-11 demonstrates the configuration required for NSF. In this case, OSPF is configured to operate in NSF mode.

Example 3-11 *Configuring OSPF to Operate in NSF Mode*

```
7507#config t
Enter configuration commands, one per line.  End with CNTL/Z.
7507(config)#hw-module slot 2 image disk0:rsp-pv-mz.120-27.S1.bin
7507(config)#hw-module slot 3 image disk0:rsp-pv-mz.120-27.S1.bin
7507(config)#redundancy
7507(config-red)#mode sso
7507(config-red)#exit
7507(config)#router ospf 100
7507(config-router)#nsf
```

Although the status of the router shows that it is in SSO mode, the status of the OSPF process now reflects NSF in Example 3-12.

Example 3-12 *Verifying OSPF in NSF Mode*

```
7507#show ip ospf
 Routing Process "ospf 100" with ID 10.10.10.4
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Incremental-SPF disabled
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x0
 Number of opaque AS LSA 0. Checksum Sum 0x0
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 1. 1 normal 0 stub 0 nssa
 Number of areas transit capable is 0
 External flood list length 0
 Non-Stop Forwarding enabled
    Area BACKBONE(0) (Inactive)
        Number of interfaces in this area is 5 (1 loopback)
        Area has no authentication
        SPF algorithm last executed 2w4d ago
        SPF algorithm executed 42 times
```

continues

Example 3-12 *Verifying OSPF in NSF Mode (Continued)*

```
        Area ranges are
        Number of LSA 1. Checksum Sum 0x3746
        Number of opaque link LSA 0. Checksum Sum 0x0
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
7507#
```

Besides getting a good grasp on the way the key protocols work and knowing the resiliency capability of the devices, it is also important to have a good understanding of the switching paths that an IP packet can possibly take within the router. This concept is important in the discussion of network resiliency, because it directly impacts how the hardware behaves as well as the resultant performance.

Impact of Different Switching Paths

A Cisco router can move a packet from one interface to another in various ways. These switching paths use different methods to achieve the same result, but the differences in the way they work affect how fast the router can perform its job. For obvious reasons, you want the router to work as fast as possible. However, depending on the network condition, the router might revert to the slowest method.

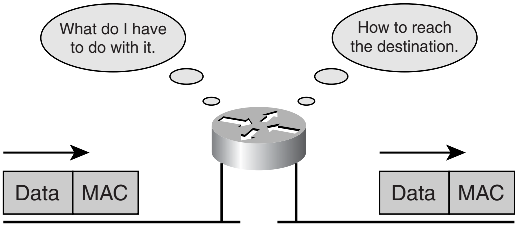
NOTE Portions of this section are adapted from "How to Choose the Best Router Switching Path for Your Network," a white paper available at Cisco.com.

To understand how these switching paths work, it helps to first understand the basic functions of the router. The role of the router is to move packets. It does it through the following steps:

1. Receive the packet.
2. Perform additional services to the packet. For example, tagging the ToS field, or changing the source or destination IP address and so on.
3. Determine how to get to the destination of the packet.
4. Determine the next hop toward the destination and which interface to use.
5. Rewrite the Media Access Control (MAC) header so that it can reach its next hop.

Figure 3-12 illustrates how a router moves packets.

Figure 3-12 *Basic Function of a Router*



Process Switching

Process switching, or sometimes referred to as punting, is the slowest among all switching paths. This is the ancestor of all routing functions, and almost all features that you see within Cisco IOS Software originate from process switching.

During process switching, the forwarding decision is based on the Routing Information Base, or RIB, and the information necessary for the MAC rewrite is taken from the ARP cache. Depending on the configuration, additional services might also be performed. However, the most significant point about process switching is that it runs as a normal process in the CPU and competes for system resources with the rest of the processes. While the CPU is handling the switching process, all other activities have to be interrupted for the job.

In process switching, the processor is heavily involved in the work of forwarding IP packets. This has the effect of slowing down the performance of the router, because every packet needs to be handled in the same manner. Most of the value-added IP services such as Network Address Translation (NAT) were introduced via the process switching method. Over time, these features have been incorporated into other methods that are more efficient and even into the ASIC.

Process switching should be avoided at all costs because it hits the control-plane function, affects performance of the router, and, in a worst-case scenario, affects the stability of the network. With process switching, a router rated as capable of forwarding millions of packets per second can be running at just thousands of packets per second. Therefore, it is important that you know which features in the Cisco IOS code will be process switched and ascertain whether there are any alternatives to the solution. Note, however, that process switching provides the most flexibility for features support. Therefore, it does have its advantages when it comes to introduction of new features in the fastest manner.

Over the years, various switching methods were devised to overcome the performance limitation of process switching, including the following:

- Interrupt context switching
- Fast Switching

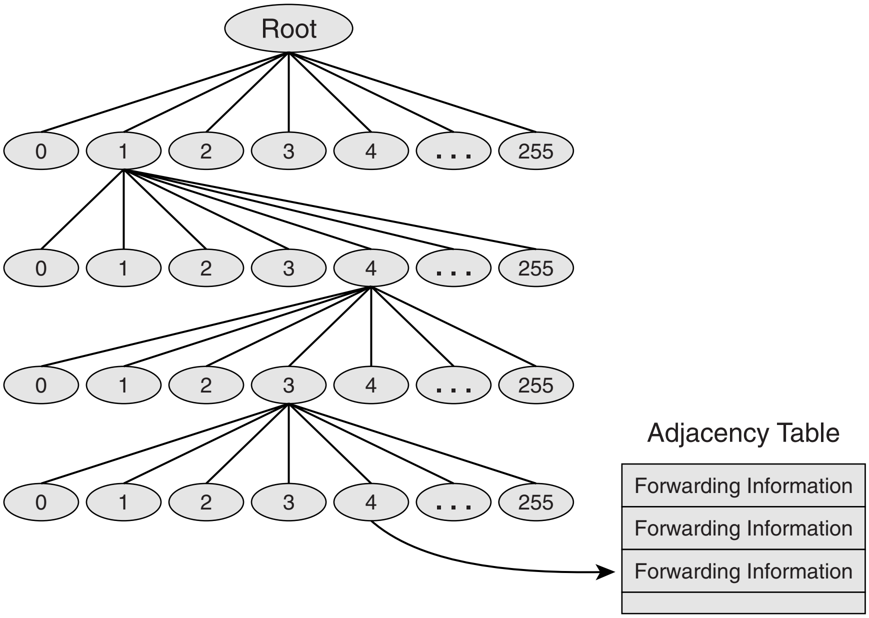
- Optimum switching
- CEF switching

Each of these switching methods stores the forwarding information in a different manner in a bid to speed up forwarding performance. In addition, each is an improvement as compared to previous ones, with CEF switching being the latest method.

Cisco Express Forwarding Switching

Cisco Express Forwarding (CEF) uses a 256-way data structure to store forwarding and MAC header rewrite information, but it does not use a tree, as do the previous methods. CEF uses a *trie*, which means the actual information being searched for is not in the data structure. Instead, the data is stored in a separate data structure, and the trie just points to it. In other words, instead of storing the outbound interface and MAC header rewrite within the tree itself, CEF stores this information in a separate data structure called the *adjacency table*, as illustrated in Figure 3-13.

Figure 3-13 *Cisco Express Forwarding (CEF)*



CEF uses a FIB to make IP destination prefix-based switching decisions. The FIB is conceptually similar to a routing table or information base. It maintains a mirror image of the forwarding information contained in the IP routing table. When routing or topology changes occur in the network, the IP routing table is updated, and those changes are reflected in the FIB. The FIB maintains next-hop address information based on the information in the IP routing table. Because there is a one-to-one correlation between FIB entries and routing table entries, the FIB contains all known routes and eliminates the need for route cache maintenance that is associated with switching paths such as fast switching and optimum switching.

Hosts in a network are said to be adjacent if they are directly connected. In addition to the FIB, CEF maintains an adjacency table to keep track of all MAC information of the adjacent hosts.

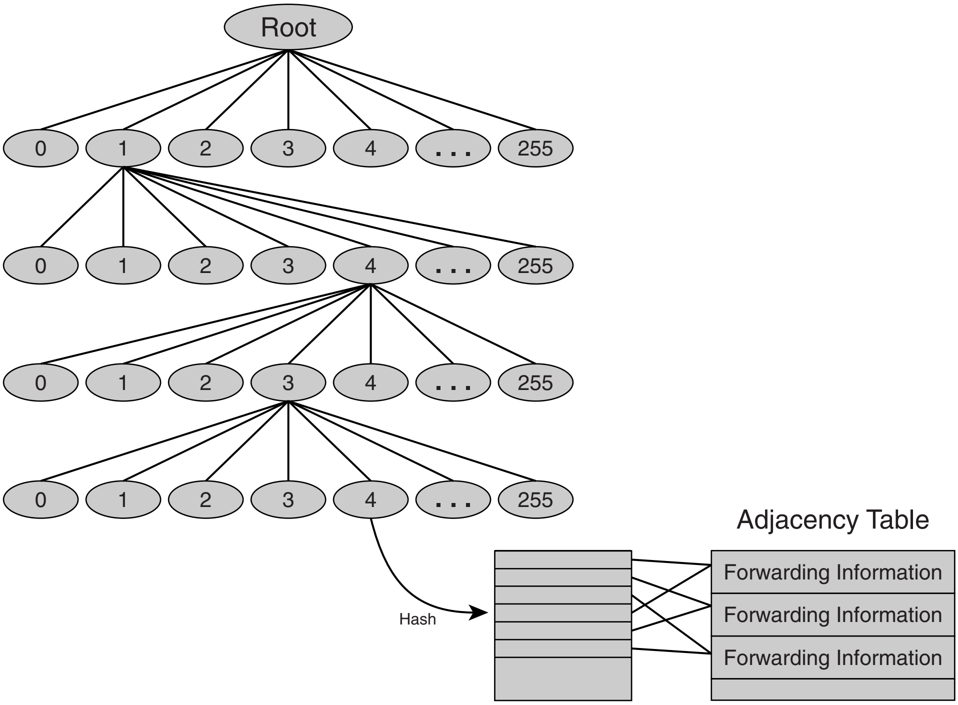
The separation of the forward information in the CEF table and the Layer 2 information in the adjacency table provides a number of benefits:

- The adjacency table can be built separately from the CEF table, allowing both to build without process switching being involved.
- The MAC header rewrite information that is used to forward a packet is not stored in the cache entries, so changes in this information do not require invalidation of cache entries.
- Recursive routes can be resolved by pointing to the recursive next hop, rather than directly to the forwarding information.
- The adjacency table can contain entries other than MAC header rewrite strings and outbound interface information. Some of the various types of entries that can be placed in the adjacency table include the following:
  - **Cache**—A MAC header rewrite string and outbound interface used to reach a particular adjacent host or router.
  - **Receive**—Packets destined to this IP address should be received by the router. This includes broadcast addresses and addresses configured on the router itself.
  - **Drop**—Packets destined to this IP address should be dropped. This could be used for traffic denied by an access list, or routed to a NULL interface.
  - **Punt**—CEF cannot switch this packet; pass it to the next-best switching method (generally fast switching) for processing.
  - **Glean**—The next hop is directly attached, but no MAC header rewrite strings are currently available.
- CEF is less CPU intensive than fast switching route caching. Therefore, more CPU resources can be dedicated to other critical activities.
- CEF offers an unprecedented level of switching consistency and stability in large dynamic networks. In a large network, fast switching cache entries are frequently invalidated because of routing changes. These changes can cause traffic to be process switched using the routing table, rather than fast switched using the route cache.

> Because the FIB lookup table contains all known routes that exist in the routing table, CEF eliminates route cache maintenance and the fast switch/process switch forwarding scenario.

CEF also takes advantage of the separation between the CEF table and the adjacency table to provide a better form of load sharing than any other switching method. An extra hash table is inserted between the CEF table and the adjacency table, as illustrated in Figure 3-14.

Figure 3-14 *CEF Load Balancing*



The CEF table points to the hash table, which contains pointers to the various adjacency table entries for available parallel paths. The source and destination addresses are passed through a hash algorithm to determine which table entry to use for each packet. Per-packet load sharing can be configured, in which case each packet uses a different hash table entry.

Each hash table has 16 entries among which the paths available are divided based on the traffic share counters in the routing table. If the traffic share counters in the routing table are all 1 (as in the case of multiple equal-cost paths), each possible next hop receives an equal number of pointers from the hash table.

With CEF, all cache aging is eliminated, and the cache is prebuilt based on the information contained in the routing table and ARP cache. Unlike process switching, there is much less processor involvement during the course of the operation. Because much less process switching is involved, the router boosts its performance, keeps its control-plane activity in check, and makes the network more robust.

Example 3-13 demonstrates the configuration for CEF.

Example 3-13 *Configuring CEF*

```
7507#config t
Enter configuration commands, one per line.  End with CNTL/Z.
7507(config)#ip cef
```

The output in Example 3-14 shows the switching status of the router.

Example 3-14 *Verifying Router Switching Status*

```
7507#show ip cef summary
IP CEF with switching (Table Version 218), flags=0x0
  18 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 1
  0 load sharing elements, 0 references
  48 in-place/0 aborted modifications
  21268 bytes allocated to the FIB table data structures
  universal per-destination load sharing algorithm, id F968AD29
  2(0) CEF resets
  Resolution Timer: Exponential (currently 1s, peak 1s)
  Tree summary:
   8-8-8-8 stride pattern
   short mask protection disabled
   18 leaves, 16 nodes using 18688 bytes
  Transient memory used: 0, max: 112
  Table epoch: 0 (18 entries at this epoch)
7507#
```

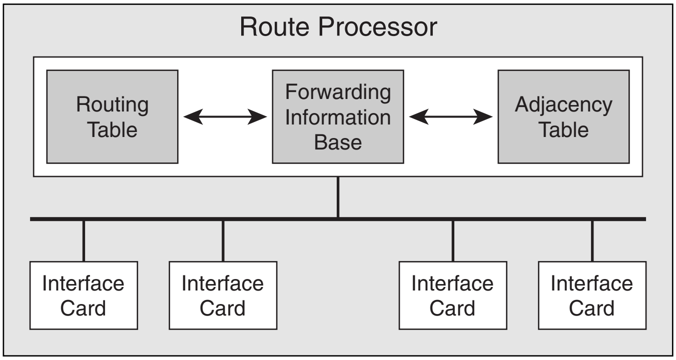
Depending on the hardware architecture, you can run CEF in two ways:

- Central CEF
- Distributed CEF

Central CEF

In central CEF mode, both the FIB and adjacency tables reside within the route processor, and the route processor performs the switching function, as illustrated in Figure 3-15.

Figure 3-15 *Central CEF Mode Operation*

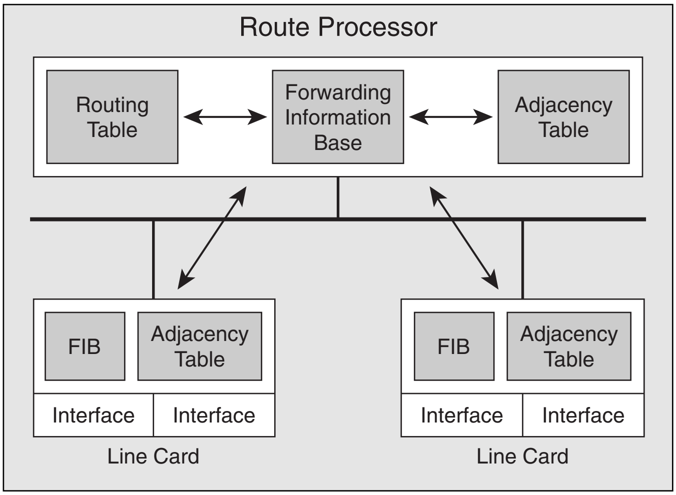


Central CEF switching mode is usually used in the lower end of the router family, because they do not support separate line cards.

Distributed CEF

In Distributed CEF (dCEF), line cards, such as the 7500 series Versatile Interface Processor (VIP) cards or Gigabit Switch Router (GSR) line cards, maintain an identical copy of the FIB and adjacency tables. The line cards perform the express forwarding between port adapters, relieving the route processor of any involvement in the switching operation. Figure 3-16 illustrates dCEF mode operation.

Figure 3-16 *Distributed CEF Mode Operation*



dCEF uses an interprocess communication (IPC) mechanism to ensure the synchronization of FIB tables and adjacency tables on the RP and line cards.

The development in the switching method not only enables a router to have better and more stable performance, it is also the basis for resiliency features such as SSO and NSF. Note that beginning from Cisco IOS Release 12.0, CEF is the preferred and default switching path, and dCEF is the only supported switching method for the Cisco 12000 series router.

Protecting the Control Plane and Data Plane

Although concepts such as control and data planes have been around for many years, not many network managers pay much attention to them—not until recently, when hackers started turning their attention to attacking the network itself.

A router, or rather, all networking devices, can be divided into three distinct components:

- **Control plane**—The brain of the hardware
- **Data plane**—The portion of the router where most of the user traffic traverses
- **Management plane**—Provides a way to configure and manage router resources

The control plane handles the well-being of the router and is responsible for activities such as routing updates, keepalives, and housekeeping of the many processes running. Recall that in CEF switching the processor has to maintain the FIB and adjacency tables. This is an example of a control-plane function. Of course, the rest of the caches, such as ARP, are also part of the function. These tables and processes that are running in the CPU of the router keep the router running properly and maintain states with the rest of the devices. If these are not properly maintained, the router will fail.

Because the control plane is such an important function, any disruption to it will have a detrimental effect. An example of an event that causes disruption of the control plane is a DoS attack. Such activity almost always causes one of the following to occur:

- Near or 100 percent CPU utilization, which inhibits the router from functioning properly
- Loss of routing protocol keepalives, which causes route flaps and network stability
- Loss of packets due to buffer exhaustion, causing dropping of legitimate IP traffic

Therefore, with more sophisticated attacks, more and more network managers are paying attention to control-plane protection. Chapter 7, “Internet Module,” discusses this topic in more detail.

The data plane handles most of the traffic-forwarding function. Recall that in dCEF switching, the line cards are forwarding traffic among the interfaces, with minimal processor intervention. This is an example of a data-plane activity. Most of the activities that happen within the data plane are value-added services such as inspection, filtering, marking, and translation. For increased performance, most of these activities are done with the help of the ASIC within the line cards, which is sometimes called a *hardware assist*. In the event that a packet is not handled by the ASIC, it will be passed to the processor for processing, which you learned about earlier in the "Process Switching" section. Any activities that cause punting negatively affect performance and so should be avoided.

The management plane provides a way to configure and manage the network. Because it can change the way the network behaves, protecting it from unlawful use is of paramount importance. The management plane plays an important role in maintaining the resiliency of the network, because it is also responsible for performance information gathering.

Routers such as the Cisco 12000 and 10000 series adopt a separate control-, data-, and management-plane design in their hardware architecture. The advantage of separating these functions, beyond dramatic performance improvements, is that the separation provides stronger fault containment. If a control plane (route processor) fails or is restarted, for instance, traffic will continue to flow through the line cards. This is the basis of how SSO and NSF work.

Establishing a Resiliency Strategy

Building a resilient IP network is not just about having redundant links and turning on the SSO features in all routers. In fact, the most difficult aspects of achieving IP resiliency is establishing an overall resiliency strategy.

With an overall resiliency strategy, you can predict what needs to be done to the network in the next few phases. You should be able to map out the logical design, knowing how it will eventually grow. From the logical design, the corresponding physical design is mapped out. Finally, both the logical and physical design translate to what sort of hardware is required, and which capacity features have to be placed in the hardware.

The strategy has to remain consistent. Many times, the resiliency of a network is compromised because of inconsistency in strategy. Shortcuts are made or different hardware is selected to do a certain task, perhaps because of shortages of funds or maybe because of changes in decision-making personnel. Problems such as these ultimately create outages later on.

Redundancy Strategy

Part of the overall resiliency strategy is how to achieve redundancy in both the logical and physical networks. Many network managers find this task challenging. For example, you might achieve physical redundancy, but because of a lack of logical redundancy, the network still experiences failure.

Logical Resiliency

When we talk about logical redundancy, we are mainly protecting important parts of the network, such as the following, from failing:

- Network paths
- Functional entity

As mentioned previously, a network path is the route that traffic traverses between a source and a destination. It is a logical entity because network paths usually arise from some route calculations (for example, a shortest-path algorithm). The determination of the path is always done by the routing protocol and the results stored in the various routers within the network. As events occur within a network (for example, when a physical link fails), the network path for a source and destination pair may potentially change. This change might result in an alternative path, or it might result in a broken connection between the same source and destination. The task then is to make sure there is always a redundant network path to an important resource within the network (to a server, for example).

Functional entity refers to the logical functions that are performed by the routers (for example, a default gateway function or a multicast routing function). A host such as a personal computer usually needs a default gateway to help it send traffic to the rest of the world. If there is only one default gateway and it fails, the personal computer will never be able to contact any other hosts except those on the same subnet. Another example is the Area Border Router (ABR) function in the OSPF network. The multiple ABRs prevent the OSPF subarea from being disconnected from area 0.

Ensuring redundancy for these logical functions is critical, because it ensures a backup in the event of a failure. Information on logical resiliency in routing is usually found in the design guides for routing protocols such as OSPF and BGP:

- *OSPF Network Design Solution*, 2nd Edition, by Tom M. Thomas. (Cisco Press, 2003. ISBN: 1587050323)
- *BGP Design and Implementation*, by Randy Zhang and Micah Bartell. (Cisco Press, 2003. ISBN: 1587051095)

Physical Resiliency

You might find the task of ensuring physical redundancy easier because it is a more visual exercise. You should look at several areas with regard to physical redundancy:

- **Device**—For device-level resiliency, look into areas such as power supply and route processors. For example, if redundant power supplies are used, they should be connected to different power sources. You also need to know how a particular device behaves under certain physical conditions such as heat and humidity. This is when certification such as Network Equipment Building System (NEBS) proves helpful.

- **Link**—For link-level redundancy, look into areas such as the number of links required and how they map to the logical design. For example, you might choose to have multiple Ethernet links between two routers. If you choose to implement EtherChannel technology, these links appear as one logical interface in the logical network. On the other hand, if these links are used individually, there will be multiple logical links in the logical design. For link redundancy, having multiple logical links might not be advantageous. For one, cost might be prohibitive, as in WAN links, or some protocols might impose a limit on the number of links that it can support.
- **Site**—With device-level and link-level resiliency addressed, the next thing to look into is whether there is a need for the entire site to be protected from disaster. This is usually applicable to data centers and for disaster recovery purposes; a remote site may be required.

Scaling Strategy

Some people might find it strange how a scaling strategy affects resiliency of a network. For one, you might not be able to tear down everything in the network just to do improvement work on a congested link. With such high expectations on the uptime on a network (remember the five-nines challenge described in Chapter 2, "Establishing a High-Availability Network"), it is almost impossible to do maintenance work without affecting network services. Therefore, many things have to be "preprovisioned" so as to avoid downtime.

As with a redundancy strategy, a scaling strategy also involves logical and physical aspects. To scale a network logically, consider aspects such as the IP addressing scheme, subnet size, and the number of subnets available within a network. You also need to look at how the routing design scales. For example, consider how many routers should be within an OSPF area, how many subnets should belong to a specific area, how many areas the network should be, and how many ABRs your network needs.

In the physical aspect, look into areas such as scaling a link speed. For example, you must decide whether to scale a 1-Gb backbone link by adding another 1-Gb link or by upgrading to an OC-48 link. The first option is called *scaling horizontally*; the latter is called *scaling vertically.* The correct choice depends on resource availability.

You might also look at things such as interface capacity, or so-called real estate, and router performance. In a chassis-based router, the number of slots, and thus the number of ports that it can support, dictates how large a network it can connect to in terms of number of links. In addition, the performance of the same router, in both switching capacity and forwarding capability, affects how much traffic it can carry at any one time.

By relying on features such as OIR, you can keep adding interfaces to a router and grow the network without affecting the rest. However, you can do so only if the router has enough slots in the first place. Therefore, having a capacity-planning exercise is important, and right sizing has to be done on the hardware. You do not want too large a router that costs a lot of money, nor do you want to run out of slots on a router.

Failure in this area usually results in network congestion and costly downtime.

Key Principles for Designing Resilient Networks

Designing a resilient IP network depends on a multitude of variables such as resources, constraints, and funds. You might not be able to get a definite list of rules for network design, but a few principles are worth noting, including the following:

- Simplicity
- Modularity
- Security

These principles are tried and tested and have been proven to be effective in every successful network.

Simplicity

In IP network design, you might be able to achieve more by having less.

One of the common mistakes many network managers make when it comes to network resiliency is an "overdesigned" network. By provisioning more redundant components than required, a network manager might actually complicate things and make matters worse. The network might become too complex to implement, monitor, and troubleshoot. The result: The network suffers from more outages.

Simplicity is perhaps the most important principle of all. It advocates implementing the minimal hardware and features to achieve the desired result, and, thus, it ultimately drives cost-savings. However, network managers must tread a fine line, because excessive cost-savings often leads to running too many logical functions within a single box. Remember, there has to be a limit to cost cutting.

Modularity

If you view the entire network as one entity, it is a huge and complex system. The network, however, is more manageable if you break it into various smaller components, as shown in Figure 3-17.

Figure 3-17 *A Modular Network*

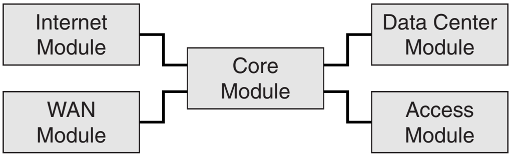


Figure 3-17 shows how a network can be broken into smaller modules, as follows:

- **Core**—The module that links the rest of the modules (commonly known as the backbone)
- **Access**—The module from which users interface with the network
- **Internet**—The module that connects the entire network to the Internet
- **WAN**—The module that connects remote branches
- **Data center**—The module that connects all the servers

Each of these smaller modules has its distinct role within the network. Because each of these modules has a unique function to perform, each requires different features to be effective in its work. For example, the features that you look for in the core module (mainly IP routing) differ from those in the access module (mainly Ethernet switching). This enables one to identify the suitable hardware to be deployed within the different modules.

In addition, it is clear that certain features may be important in a module but may not be so in another. Often, network managers ask for every feature on a particular piece of hardware, only to pay too much for a piece of hardware that eventually does little. The excuse for doing this is always the "just in case" mentality. However, with proper logical design in place, and most important, a consistent strategy, you can avoid this.

Another advantage of having a modular network design is it enables you to isolate problems within a module. If you contain the problem, the rest of the network can then continue to function, which means fewer users are affected and more overall uptime of the network.

Security

With hacking tools readily available now, it no longer takes a professional to do damage to your network. A youth sitting in his study room with broadband access to the Internet is all it takes. Therefore, security is paramount and should not be compromised when designing the network.

The job of the network is to transport IP packets from source to destination. Therefore, the resources required to perform the job, the routing information and the devices, become important entities and natural targets of attacks. Protecting these resources becomes a paramount task when designing the network. These resources can be exploited in many ways; the best source of information is a white paper under the Cisco SAFE Blueprint series:

> SAFE: Best Practices for Securing Routing Protocols, http://www.cisco.com/en/US/netsol/ns340/ns394/ns171/ns128/networking_solutions_white_paper09186a008020b51d.shtml

Securing the routing infrastructure is just one aspect of security. You should also look into many other areas, such as IP telephony, wireless network, as well as mitigating worm and virus attacks. A good reference to start with is the Cisco SAFE Blueprint, which you can find at the following URL:

> http://www.cisco.com/en/US/netsol/ns340/ns394/ns171/ns128/networking_solutions_package.html

Summary

Building a resilient IP network requires you to thoroughly understand the way the three protocols—IP, TCP, and UDP—behave. It is also important to understand the factors that affect the performance of the network. Besides the software, understanding the high-availability feature of the hardware, and its impact on these protocols, helps you to choose the right device to build the network. However, the first step to a resilient network is to have a clear strategy as to how it should be achieved, bearing in mind that simplicity, modularity, and security play important roles.

INDEX