CISCO

IP Routing on Cisco IOS®, IOS XE, and IOS XR

An Essential Guide to Understanding and Implementing IP Routing Protocols

Brad Edgeworth, CCIE No. 31574
Aaron Foss, CCIE No. 18761
Ramiro Garza Rios, CCIE No. 15469

ciscopress.com

IP Routing on Cisco IOS, IOS XE, and IOS XR

An Essential Guide to Understanding and Implementing IP Routing Protocols

Brad Edgeworth, CCIE No. 31574
Aaron Foss, CCIE No.18761
Ramiro Garza Rios, CCIE No. 15469

Cisco Press

800 East 96th Street

Indianapolis, IN 46240

IP Routing on Cisco IOS, IOS XE, and IOS XR

Brad Edgeworth, Aaron Foss, Ramiro Garza Rios

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

Printed in the United States of America

First Printing December 2014

Library of Congress Control Number: 2014957562

ISBN-13: 978-1-58714-423-3

ISBN-10: 1-58714-423-9

Warning and Disclaimer

This book is designed to provide information about Cisco IOS, IOS XE, and IOS XR. 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 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 book or from the use of the discs or programs that may accompany it.

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

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 email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Trademark Acknowledgments

Publisher: Paul Boger

Associate Publisher: Dave Dusthimer

Business Operation Manager, Cisco Press: Jan Cornelssen

Acquisitions Editor: Denise Lincoln

Managing Editor: Sandra Schroeder

Senior Development Editor: Christopher Cleveland

Project Editor: Seth Kerney

Copy Editor: Keith Cline

Technical Editors: Richard Furr, Pete Lumbis

Editorial Assistant: Vanessa Evans

Book Designer: Gary Adair

Cover Designer: Mark Shirar

Composition: Trina Wurst

Indexer: Heather McNeill

Proofreader: Apostrophe Editing Services

About the Authors

Brad Edgeworth, CCIE No. 31574 (R&S & SP), has been with Cisco since 2011, working as a Systems Engineer and a Technical Leader. Brad is a distinguished speaker at Cisco Live, where he has presented on IOS XR. Before joining Cisco, Brad worked as a network architect and consultant for various Fortune 500 companies. Brad's other certifications include Cisco Certified Design Professional (CCDP) and Microsoft Certified Systems Engineer (MCSE). Brad has been working in the IT field for the past 18 years, with an emphasis on enterprise and service provider environments from an architectural and operational perspective. Brad holds a bachelor of arts degree in computer systems management from St. Edward's University in Austin, Texas.

Aaron Foss, CCIE No. 18761 (R&S & SP), is a High Touch Engineer with Cisco's Focused Technical Support (FTS) organization. He works with large service providers to troubleshoot issues relating to Multiprotocol Label Switching (MPLS), quality of service (QoS), and IP routing protocols. Aaron has more than 15 years of experience designing, deploying, and troubleshooting IP networks. He holds a bachelor of science degree in management information systems from Rochester Institute of Technology.

Ramiro Garza Rios, CCIE No. 15469 (R&S, SP, and Security), is a Senior Network Consulting Engineer with Cisco Advanced Services. His current role consists of planning, designing, implementing, and optimizing next-generation (NGN) service provider networks in the United States. He has been with Cisco for more than 8 years and has 14 years of networking industry experience. Before joining Cisco, Ramiro was a Network Consulting and Presales Engineer for a Cisco Gold Partner in Mexico, where he was involved in the planning, design, and implementation of many enterprise and service provider networks. He holds a bachelor of science degree in electronic engineering from the Instituto Tecnologico de Reynosa and lives with his wife and four children in Cary, North Carolina.

About the Technical Reviewers

Richard Furr, CCIE No.9173 (R&S & SP), is a Technical Leader with Cisco's Technical Assistance Center (TAC). For the last 13 years, Richard has worked for Cisco TAC and High Touch Technical Support (HTTS) organizations, supporting service providers and large enterprise environments with a focus on troubleshooting routing protocols, MPLS, IP multicast and QoS.

Pete Lumbis, CCIE No. 28677 (R&S) and CCDE 20120003, is an expert in routing technologies including Border Gateway Protocol (BGP), MPLS, and multicast. He spent five years working in the Cisco TAC as the Routing Protocols Escalation Engineer supporting all of Cisco's customers. Most recently, Pete is focused on network design and architecture at Microsoft. Pete has been a distinguished speaker at Cisco Live on routing fast convergence and IOS routing internals.

Dedications

This book is dedicated to my loving wife Tanya, who has endured and supported me through all my endeavors.

-Brad

I would like to dedicate this book to my supportive wife, Anne, and to my children, Ashley, Benny, and Clara, for giving up some weekend time with Dad so that I could write this book.

-Aaron

I would like to dedicate this book to my wonderful and beautiful wife, Mariana, and to my children Ramiro, Frinee, Felix, and Lucia for their love, patience, sacrifice, and support while writing this book.

To my parents, Ramiro Garza and Blanca Dolores Rios, for their continued support, love, encouragement, guidance, and wisdom.

And most importantly, I would like to thank God for all His blessings in my life.

-Ramiro

Acknowledgments

Brad Edgeworth:

A special thank you goes to Norm Dunn, Jocelyn Lau, Brett Bartow, and Denise Lincoln for making this book possible.

A debt of gratitude goes to my co-authors, Aaron and Ramiro. You accepted the challenge of helping me write this book. Little did you know that this project would become your second job. Some of the book's best chapters were supposed to be small, but exploded in to mini-novels to cover the topic properly. Your knowledge and dedication to this project are appreciated more than you will ever know.

To our technical editors, Richard and Pete: Thank you for finding all of our mistakes. In addition to your technical accuracy, your insight into the technologies needed by Cisco customers versus crazy ninja router tricks has kept the size of the book manageable and the content relevant.

Aaron, Ramiro, and I want to thank the Cisco Press team for their assistance and insight throughout this project. Chris Cleveland, you have been a pleasure to work with, and your attention to detail is simply amazing. It has been an educational experience for the three of us.

A special thanks to the Cisco HTTS RP and IOS XR teams, who continuously educate those about routing protocols. A special recognition to Hunter, Yigal, and Jimmy—you guys are rock stars!

Many people within Cisco have provided feedback and suggestions to make this a great book. Thanks to all who have helped in the process, especially Umair Arshad, Heather Bunch, Luc de Ghein, David Roehsler, Faraz Shamim, Craig Smith, and Mobeen Tahir.

Aaron Foss:

I would like to thank my co-authors Brad and Ramiro for their amazing collaboration on this project. Brad, you have an extraordinary determination and drive that I admire greatly; and Ramiro, your technical knowledge and ability to make us laugh throughout the process of writing this book was much appreciated.

Finally, I want to acknowledge my manager, Zulfiqar Ahmed, for supporting me and encouraging me to undertake this book endeavor.

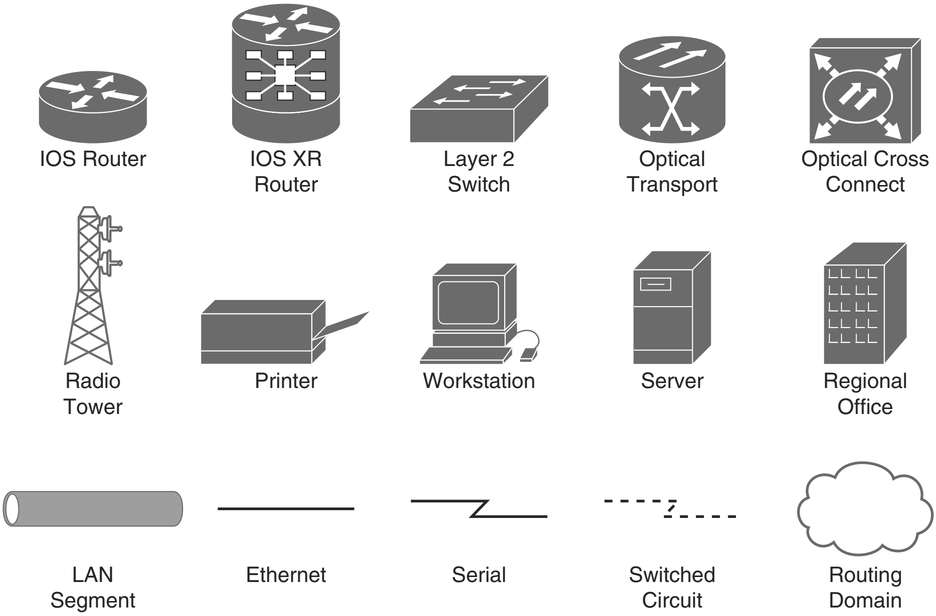
Ramiro Garza Rios:

I would like to thank God for giving me the opportunity to work on this book. I would like to acknowledge my co-author Brad for the inception of this book and for being persistent until it became a reality. I would also like to acknowledge both of my co-authors, Aaron and Brad, for the great teamwork, dedication, and valuable input provided throughout the project.

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 Cisco's Command Reference. The Command Reference describes these conventions as follows:

- Boldface indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a show command).
- 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.

Note This book covers multiple operating systems, and a differentiation of icons and router names indicate the appropriate operating system that is being referenced. IOS and IOS XE use router names like R1 and R2 and are referenced by the IOS router icon. IOS XR routers will use router names like XR1 and XR2 and are referenced by the IOS XR router icon.

Foreword

Service providers and even large, well-established enterprises, while they continue to sweat some legacy networking assets, they also realize the operational efficiencies gained by converging these disparate assets onto a common IP infrastructure. Furthermore, they generally understand the benefits of being able to offer new and innovative services with quicker time-to-market deployment with one unified converged IP backbone. Many service providers and enterprises have built out new IP backbones and are already realizing benefits of converged networking, but many have not, plus most have not realized the full potential of capability and revenue generation they can provide. This is where the need and demand for highly skilled IP network engineers becomes critical to the evolution of these IP network infrastructures, and where learning products such as Cisco career certifications and this Cisco Press resource shines in value.

This Cisco Press book is an excellent self-study resource to help aid candidates in preparing to pass exams associated with the CCNA Service Provider, CCNP Service Provider, and CCIE Service Provider career certifications. Exams associated with these Cisco certifications cover technology areas such as routing protocols (Enhanced Interior Gateway Routing Protocol [EIGRP], Open Shortest Path First [OSPF] Protocol, Intermediate System-to-Intermediate System [IS-IS] Protocol, and Border Gateway Protocol [BGP]), multicast, IPv6, and high availability. This book serves as a valuable aid in preparation in these areas. Furthermore, the book covers these topics across multiple Cisco operating system implementations, such as Cisco IOS and IOS XR, which are also covered within the noted Cisco career certifications. This resource can also aid in prepping candidates pursuing CCNA-CCNP-CCIE Routing and Switching certifications. Lastly, this book is valuable in general for learners looking to simply increase their technical understanding about how to configure routing protocols, multicast, IPv6, and high availability.

We hope and expect you'll find this book to be a valuable and frequently referenced technical aid, and a unique reference book for your personal library.

Norm Dunn

Senior Product Manager, Learning@Cisco

Global Product Management, Service Provider Portfolio

Cisco Systems, Inc.

Introduction

Within Cisco's Focused Technical Support (FTS) organization, a large number of questions about the IOS, IOS XE, and IOS XR operating systems are encountered on a daily basis. This book answers IP routing questions, in addition to covering the implementation and troubleshooting differences between the operating systems.

In alignment with the saying "a picture is worth a thousand words," multiple illustrations are included in the chapters to explain the various concepts. All protocols are presented conceptually, with applicable illustrations, configurations, and appropriate output. The scope of this book evolved to include the IOS and IOS XE operating systems so that non-IOS XR users could benefit from the explanations on the routing protocols. The books structure explains a concept, and then provides the configuration commands and verification of the feature in small, digestible nuggets of information.

This book's content was created in alignment with Learning@Cisco to address the demand for more efficient self-study content for the Cisco Career Service Provider Certifications.

This book encompasses content spread across multiple sources and presents them in a different perspective while covering updated standards and features that are found in enterprise and service provider networks.

Who Should Read This Book?

Network engineers, consultant, and students who want to understand the concepts and theory of EIGRP, OSPF, IS-IS, BGP, and multicast routing protocols on Cisco IOS, IOS XE, and IOS XR operating systems should read this book.

The book's content is relevant to network engineers in various stages of their career and knowledge. Every topic assumes minimal knowledge and explains the protocol from a ground-up perspective. For the advanced network engineers, relevant information on the routing protocol behavior is included. Differences in protocol behavior between IOS, IOS XE, and IOS XR are explicitly identified for each protocol.

How This Book Is Organized

Although this book could be read cover to cover, it is designed to be flexible and allow you to easily move between chapters and sections of chapters to cover just the material that you need more work with. This book is organized into seven distinct sections.

Part I of the book provides a brief review of the operating systems, IP addressing, and networking fundamentals.

- **Chapter 1, "Introduction to the Operating Systems:"** This chapter provides a high-level comparison of the network operating system architectures. An overview of the CLI configuration is provided so that users are comfortable with logging in and configuring the routers.

- **Chapter 2, "IPv4 Addressing:"** This chapter explains the IPv4 addressing structure, the need for subnetting, and the techniques to differentiate a network address from a host address.
- **Chapter 3, "How a Router Works:"** This chapter explains the reasons for using a routing protocol, the types of routing protocols, and the logic a router uses for forwarding packets.

Part II of the book explains static routing, EIGRP, OSPF, IS-IS, and BGP routing protocols.

- **Chapter 4, "Static Routes:"** This chapter explains connected networks and static routes from the perspective of a router.
- **Chapter 5, "EIGRP:"** This chapter explains the EIGRP routing protocol and how distance vector routing protocols work.
- **Chapter 6, "OSPF:"** This chapter explains the basic fundamentals of the routing protocol, and its operational characteristics.
- **Chapter 7, "Advanced OSPF:"** This chapter explains the reason for breaking an OSPF routing domain into multiple areas, techniques for optimization, and how to determine the best path.
- **Chapter 8, "IS-IS:"** This chapter explains the history of the IS-IS routing protocol, along with the similarities and differences it has with OSPF.
- **Chapter 9, "Advanced IS-IS:"** This chapter explains multilevel routing in an IS-IS domain, optimization techniques, and the path selection process.
- **Chapter 10, "Border Gateway Protocol:"** This chapter explains the fundamental concepts of BGP sessions and route advertisement. The chapter covers the differences between external and internal peers.

Part III of the book explains the advanced routing concepts that involve routing policies and redistribution.

- **Chapter 11, "Route Maps and Route Policy Language:"** This chapter explains prerequisite concepts such as matching networks prefixes with an access control list (ACL), prefix list or BGP advertisements with regex queries. This chapter also explains how IOS and IOS XE route maps can manipulate traffic. The chapter then discusses how IOS XR's route policy language was designed to provide clarity and scalability.
- **Chapter 12, "Advanced Route Manipulation:"** This chapter discusses policy-based routing, along with administrative distance manipulation, to modify route forwarding behavior. The chapter concludes by describing how to filter out specific routes from routing protocol participation.
- **Chapter 13, "Route Redistribution:"** This chapter explains the ability to inject network prefixes learned from one routing protocol into another routing protocol. The chapter provides a thorough coverage on the rules of redistribution, problems associated with mutual redistribution, and methods for remediation.

Part IV of the book revisits BGP and describes how prefix lists, route maps, route policies, and redistribution can be used for traffic engineering.

- **Chapter 14, "Advanced BGP:"** BGP communities, summarizations, and other router conservation techniques are explained in this chapter.
- **Chapter 15, "BGP Best Path Selection:"** This chapter provides a through explanation of the best path selection algorithm and the ramifications that the selection has for other routers in the autonomous system. BGP route reflectors are examined, along with suboptimal routing due to path information loss. The chapter concludes with an overview of the various techniques available to optimize traffic flows when using route reflectors.

Part V of the book explains multicast traffic, the benefits of multicast, and configuration.

- **Chapter 16, "IPv4 Multicast Routing:"** This chapter describes the benefits of multicast. Key multicast features such as Internet Group Management Protocol (IGMP), Protocol Independent Multicast (PIM), rendezvous points, multicast distribution trees are all discussed.
- **Chapter 17, "Advanced IPv4 Multicast Routing:"** Large multicast networks require additional features to provide scalability and reachability between routing domains and autonomous systems. This chapter explains the advanced features: Multicast Source Discovery Protocol (MSDP), Source Specific Multicast (SSM), multicast boundaries, and multicast BGP.

Part VI of the book explains the IPv6 address structure, the changes to the routing protocols, and IPv6 multicast routing.

- **Chapter 18, "IPv6 Addressing:"** This chapter describes the IPv6 address structure. The protocol stack's neighbor discovery mechanisms are outlined, such as router advertisement messages, stateless address autoconfiguration, and duplicate address detection.
- **Chapter 19, "IPv6 Routing:"** This chapter outlines the subtle command structure and protocol mechanics changes between the IPv4 and IPv6 routing protocols.
- **Chapter 20, "IPv6 Multicast Routing:"** This chapter explains the fundamental differences between IPv4 and IPv6 multicast routing while emphasizing technologies like Multicast Listener Discovery (MLD), SSM, Embedded RP, and multicast boundaries.

Part VII, which can be found online at this book's site, explains the concepts involved with improving the operational uptime of the network.

- **Chapter 21, "High Availability:"** This chapter describes the techniques available to improve network availability and provide fast routing convergence.

Final Words

This book is an excellent self-study resource to learn the routing protocols on Cisco IOS, IOS XE, and IOS XR operating systems. However, reading is not enough, and anyone who has obtained their CCIE will tell you that you must implement a technology to fully understand it. Our topologies are intentionally kept small to explain the routing concepts. We encourage the reader to re-create the topologies and follow along with the examples. A variety of resources are available that will allow you to practice the same concepts. Look online for the following:

- Online simulators at Learning@Cisco
- Online rack rentals
- Free demo versions of Cisco CSR 1000V (IOS XE)
- Free demo versions of Cisco IOS XRv (IOS XR)

Happy labbing!

This page intentionally left blank

Chapter 3

How a Router Works

This chapter covers the following topics:

- IP routing
- IP packet switching
- Planes of operation

The previous chapters described that a router is necessary to transmit packets between network segments. This chapter explains the process a router uses to accomplish this task. By the end of this chapter, you should have a good understanding of how a router performs IP routing and IP packet forwarding between different network segments.

IP Routing

A router's primary function is to move an IP packet from one network to a different network. A router learns about nonattached networks through static configuration or through dynamic IP routing protocols.

Dynamic IP routing protocols distribute network topology information between routers and provide updates without intervention when a topology change in the network occurs. Design requirements or hardware limitations may restrict IP routing to static routes, which do not accommodate topology changes very well, and can burden network engineers depending on the size of the network. Routers try to select the best loop-free path in a network that forwards a packet to its destination IP address.

A network of interconnected routers and related systems managed under a common network administration is known as an *autonomous system*. The Internet is composed of thousands of autonomous systems spanning the globe.

The common dynamic routing protocols found in networks today are as follows:

- RIPv2 (Routing Information Protocol Version 2)
- EIGRP (Enhanced Interior Gateway Routing)
- OSPF (Open Shortest Path First) Protocol
- IS-IS (Intermediate System-to-Intermediate System) Protocol
- BGP (Border Gateway Protocol)

With the exception of BGP, the protocols in the preceding list are designed and optimized for routing within an autonomous system and are known as *internal gateway protocols* (IGPs). External gateway protocols (EGPs) route between autonomous systems. BGP is an EGP protocol but can also be used within an autonomous system. If BGP exchanges routes within an autonomous system, it is known as an *internal BGP* (iBGP) session. If it exchanges routes between different autonomous systems, it is known as an *external BGP* (eBGP) session.

Figure 3-1 shows an illustration of how one or many IGPs as well as iBGP can be running within an autonomous system and how eBGP sessions interconnect the various autonomous systems together.

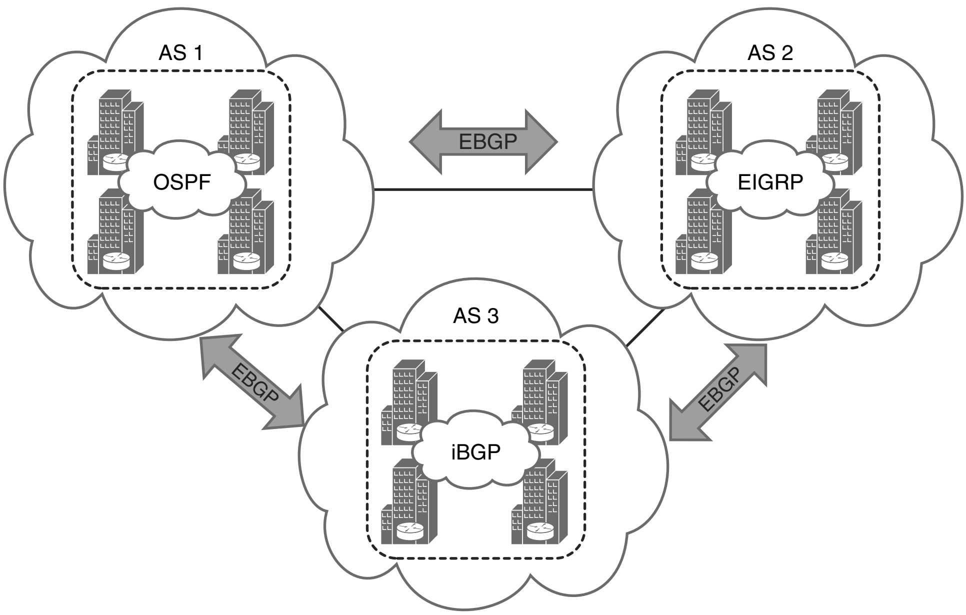


Figure 3-1 *Autonomous Systems and How They Interconnect*

EGPs and IGPs use different algorithms for path selection and are discussed in the following sections.

Distance Vector Algorithms

Distance vector routing protocols, such as RIP, advertise routes as vectors (distance, vector), where distance is a metric (or cost) such as hop count and vector is the next-hop router's IP used to reach the destination:

- **Distance:** The distance is the route metric to reach the network.
- **Vector:** The vector is the interface or direction to reach the network.

When a router receives routing information from a neighbor, it stores it in a local routing database as it is received and the distance vector algorithm (also known as *Bellman-Ford* and *Ford-Fulkerson* algorithms) is used to determine which paths are the best loop-free paths to each reachable destination. Once the best paths are determined, they are installed into the routing table and are advertised to each neighbor router.

Routers running distance vector protocols advertise the routing information to their neighbors from their own perspective, modified from the original route that it received. For this reason, distance vector protocols do not have a complete map of the whole network; instead, their database reflects that a neighbor router knows how to reach the destination network and how far the neighbor router is from the destination network. They do not know how many other routers are in the path toward any of those networks. The advantage of distance vector protocols is that they require less CPU and memory and can run on low-end routers.

An analogy commonly used to describe distance vector protocols is that of a road sign at an intersection that indicates the destination is 20 miles to the west; this information is trusted and blindly followed, without really knowing whether there is a shorter or better way to the destination or if the sign is even correct. Figure 3-2 illustrates how a router using a distance vector protocol views the network and the direction that R3 needs to go to reach the 192.168.1.0/24 subnet.

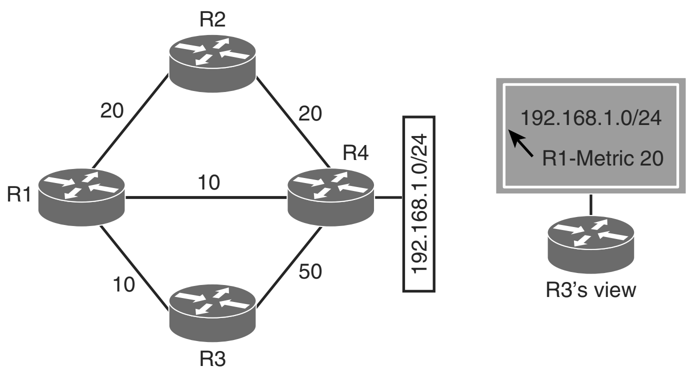


Figure 3-2 *Distance Vector Protocol View of the Network*

Enhanced Distance Vector Algorithm

The Diffused Update Algorithm (DUAL) is an enhanced distance vector algorithm that EIGRP uses to calculate the shortest path to a destination within a network. EIGRP advertises network information to its neighbors as other distance vector protocols do, but it has some enhancements as its name suggests. Some of the enhancements introduced into this algorithm compared to other distance vector algorithms are the following:

- Rapid convergence time for changes in the network topology.
- Only sends updates when there is a change in the network. It does not send full routing table updates in a periodic fashion like distance vector protocols.
- It uses hellos and forms neighbor relationships just like link-state protocols.
- It uses bandwidth, delay, reliability, load, and maximum transmission unit (MTU) size instead of hop count for path calculations.
- It has the option to load balance traffic across equal or unequal metric cost paths.

EIGRP is sometimes referred to as a *hybrid routing protocol* because it has characteristics of both distance vector and link-state protocols, as shown in the preceding list (for example, forming adjacencies with neighbor routers and relying on more advanced metrics such as bandwidth other than hop count for its best path calculations).

Link-State Algorithms

Link-state dynamic IP routing protocols advertise the link state and link metric for each of their connected links and directly connected routers to every router in the network. OSPF and IS-IS are two common link-state routing protocols found in enterprise and service provider networks. OSPF advertisements are called *link-state advertisements* (LSAs), and IS-IS uses link-state packets (LSPs) for its advertisements.

As a router receives an advertisement from a neighbor, it stores the information in a local database called the *link-state database* (LSDB), and advertises the link-state information on to each of its neighbor routers exactly as it was received. The link-state information is essentially flooded throughout the network from router to router unchanged, just as the originating router advertised it. This allows all the routers in the network to have a synchronized and identical map of the network.

Using the complete map of the network, every router in the network then runs the Dijskstra shortest path first (SPF) algorithm (developed by Edsger W. Dijkstra) to calculate the best shortest loop-free paths. The link-state algorithm then populates the routing table with this information.

Due to having the complete map of the network, link-state protocols usually require more CPU and memory than distance vector protocols, but they are less prone to routing loops and make better path decisions. In addition, link-state protocols are equipped with extended capabilities such as opaque LSAs for OSPF and TLVs (type/length/value) for IS-IS that allows them to support features commonly used by service providers such as MPLS traffic engineering.

An analogy for link-state protocols is a GPS navigation system. The GPS navigation system has a complete map and can make the best decision as to which way is the shortest and best path to reach the destination. Figure 3-3 illustrates how R3 would view the network to reach the 192.168.1.0/24 subnet.

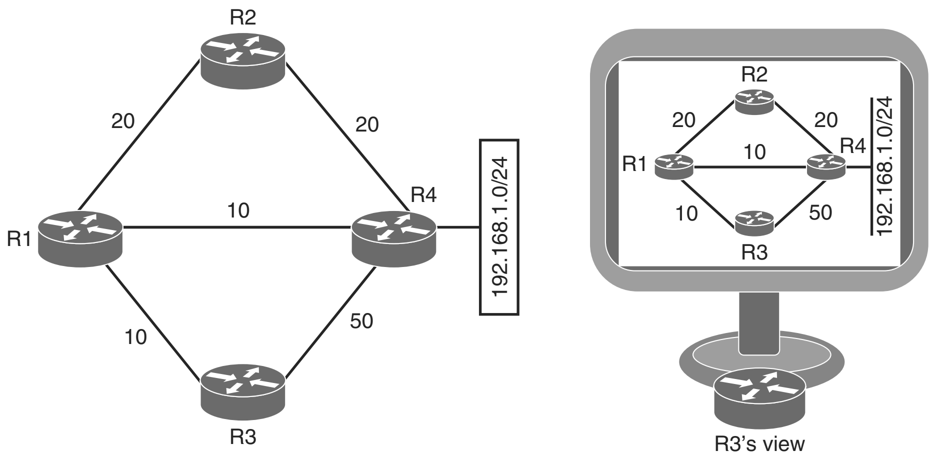


Figure 3-3 *Link-State Protocol View of the Network*

Path Vector Algorithm

A path vector protocol such as BGP is similar to a distance vector protocol; the difference is that instead of looking at the distance to determine the best loop-free path, it looks at various BGP path attributes. BGP path attributes include autonomous system path (AS_Path), Multi-Exit Discriminator (MED), origin, next hop, local preference, atomic aggregate, and aggregator. BGP path attributes are covered in Chapter 10, "BGP," and Chapter 14, "Advanced BGP."

A path vector protocol guarantees loop-free paths by keeping a record of each autonomous system that the routing advertisement traverses. Any time a router receives an advertisement in which it is already part of the autonomous system path, the advertisement is rejected because accepting the autonomous system path would effectively result in a routing loop.

Figure 3-4 illustrates this concept where autonomous system 1 advertises the 10.1.1.0/24 network to autonomous system 2. Autonomous system 2 receives this information and adds itself to the autonomous system path and advertises it to autonomous system 4. Autonomous system 4 adds itself to the path and advertises it to autonomous system 3. Autonomous system 3 receives the route advertisement and adds itself to the path as well. However, when autonomous system 3 advertises that it can reach 10.1.1.0/24 to autonomous system 1, autonomous system 1 discards the advertisement because the autonomous system path (path vector) contained in the advertisement includes its autonomous system number (autonomous system 1). When autonomous system 3 attempts to advertise reachability for 10.1.1.0/24 to autonomous system 2, autonomous system 2 also discards it because the advertisement includes autonomous system 2 in the autonomous system path, too.

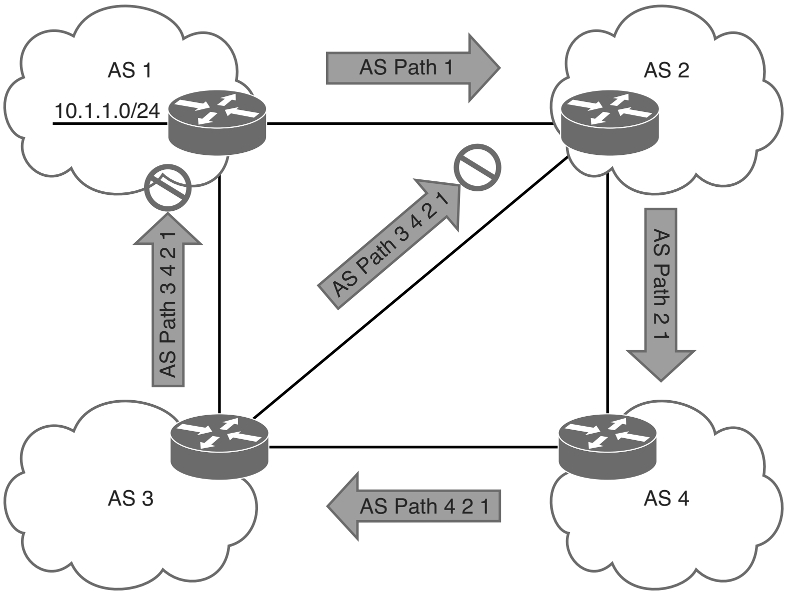


Figure 3-4 *Path Vector Loop Avoidance*

All BGP path attributes and how to manipulate them to influence the best path selection process are covered in Chapter 15, "BGP Best Path Selection."

Routing Table

A router identifies the path a packet should take by evaluating the following components on a router:

- **Prefix length:** The prefix length represents the number of leading binary bits in the subnet mask that are in the on position.
- **Administrative distance:** Administrative distance (AD) is a rating of the trustworthiness of a routing information source. If a router learns about a route to a destination from more than one routing protocol and they all have the same prefix length, AD is compared. The preference is given to the route with the lower AD.
- **Metrics:** A unit of measure used by a routing protocol in the best path calculation.

Prefix Length

Let's look at a scenario of a router selecting a route when the packet destination is within the network range for multiple routes. Assume that a router has the following routes with various prefix lengths in the routing table:

- 10.0.3.0/28
- 10.0.3.0/26
- 10.0.3.0/24

Because each of these routes, also known as *prefix routes* or simply *prefixes*, has a different prefix length (subnet mask), they are considered to be different destinations, and they will all be installed into the routing table. This is represented in Table 3-1.

Table 3-1 *Representation of Routing Table*

Prefix	Subnet Range	Next Hop	Outgoing Interface
10.0.3.0/28	10.0.3.0 – 10.0.3.15	10.1.1.1	Gigabit Ethernet 1/1
10.0.3.0/26	10.0.3.0 – 10.0.3.63	10.2.2.2	Gigabit Ethernet 2/2
10.0.3.0/24	10.0.3.0 – 10.0.3.255	10.3.3.3	Gigabit Ethernet 3/3

If a packet needs to be forwarded, the route chosen depends on the prefix length, where the *longest prefix length* is always preferred. For example, /28 is preferred over /26, and /26 is preferred over /24. The following is an example using Table 3-1 as a reference:

- If a packet needs to be forwarded to 10.0.3.14, it would match all three routes, but it would be sent to next hop 10.1.1.1 and outgoing interface Gigabit Ethernet 1/1 because 10.0.3.0/28 has the longest prefix match.
- If a packet needs to be forwarded to 10.0.3.42, it would match 10.0.3.0/24 and 10.0.3.0/26, so the packet would be sent to 10.2.2.2 and outgoing interface Gigabit Ethernet 2/2 because 10.0.3.0/26 has the longest prefix match.
- If a packet needs to be forwarded to 10.0.3.100, it matches only 10.0.3.0/24, so the packet is sent to 10.3.3.3 and outgoing interface Gigabit Ethernet 3/3.

Administrative Distance

As each routing protocol receives updates and other routing information, it chooses the best path to any given destination and attempts to install this path into the routing table. Table 3-2 provides the default AD for the routing protocols covered in this book.

Table 3-2 *Routing Protocol Default Administrative Distances*

Routing Protocol	Default Administrative Distance
Connected	0
Static	1
eBGP	20
EIGRP summary route	5
EIGRP (internal)	90
OSPF	110
IS-IS	115
RIP	120
EIGRP (external)	170
iBGP	200

For example, if OSPF learns of a best path toward 10.0.1.0/24, it first checks to see whether an entry exists in the routing table. If it does not exist, the route is installed into the Routing Information Base (RIB). If the route already exists in the RIB, the router decides whether to install the route presented by OSPF based on the AD of the route in OSPF and the AD of the existing route in the RIB. If this route has the *lowest AD* to the destination (when compared to the other route in the table), it is installed in the routing table. If this route is not the route with the best AD, the route is rejected.

Consider another example on this topic. A router has OSPF, IS-IS, and EIGRP running, and all three protocols have learned of the destination 10.3.3.0/24 network with a different best path and metric.

Each of these three protocols will then attempt to install the route to 10.0.3.0/24 into the routing table. Because the prefix length is the same, the next decision point is the AD, where the routing protocol with the lowest AD installs the route into the routing table.

Because the EIGRP internal route has the best AD, it is the one installed into the routing table:

10.0.3.0/24	**EIGRP**	**90 <<< Lowest AD Installed in Route Table**
10.0.3.0/24	OSPF	110
10.0.3.0/24	IS-IS	115

The routing protocol or protocols that failed to install their route into the table (in this example, that would be OSPF and IS-IS) will hang on to this route to use it as a backup route and will tell the routing table process to report to them if the best path fails so that they can then try to reinstall this route.

For example, if the EIGRP route 10.0.3.0/24 installed in the routing table fails for some reason, the routing table process calls OSPF and IS-IS, and requests them to reinstall the route in the routing table. Out of these two protocols, the preferred route is chosen based on AD, which would be OSPF because of its lower AD.

The default AD might not always be suitable for a network; for instance, there might be a requirement to adjust it so that OSPF routes are preferred over EIGRP routes. However, changing the AD on routing protocols can have severe consequences, such as routing loops and other odd behavior in a network. It is recommended that the AD be changed only with extreme caution, and only after what needs to be accomplished has been thoroughly thought out. A good backup plan is recommended in case things do not turn out as planned.

Metrics

As discussed in the previous section, routes are chosen and installed into the routing table based on the routing protocol's AD. The routes learned from the routing protocol with the lowest AD are the ones installed into the routing table. If there are multiple paths to the same destination from a single routing protocol, these paths would have the same AD; for this case, the best path is selected within the routing protocol. Most protocols use the path with the best metric, but OSPF and IS-IS have additional logic that preempts the lowest metric.

If a routing protocol identifies multiple paths as a *best path*, and supports multiple path entries, the router installs the maximum number of paths allowed per destination. This is known as *equal-cost multipath* (ECMP) and provides load sharing across all links.

For example, Figure 3-5 illustrates a network running OSPF to reach the prefix 10.3.3.0/24. Router 1 (R1) has two equal-cost paths; therefore, it will install both in the routing table.

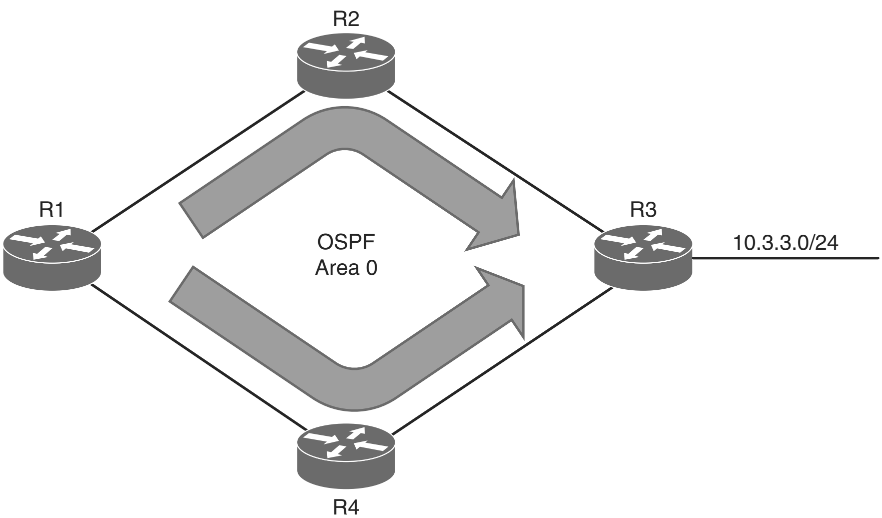


Figure 3-5 *OSPF ECMP Technology*

Example 3-1 confirms that both paths have been installed into the RIB, and because the metrics are identical, this confirms the router is using ECMP.

Example 3-1 *R1's Routing Table Showing the ECMP Paths to 10.3.3.0/24*

```
R1#show ip route
! Output omitted for brevity
O     10.3.3.0/24 [110/30] via 10.4.4.4, 00:49:12, GigabitEthernet0/0
                  [110/30] via 10.2.2.2, 00:49:51, GigabitEthernet0/2
```

Note Best path metric calculation and the default and maximum ECMP paths allowed for each routing protocol vary. This is covered in later routing protocol-related chapters.

Virtual Routing and Forwarding

Virtual Routing and Forwarding (VRF) is a technology that allows multiple independent virtual routing table and forwarding table instances to exist concurrently in a router. This can be leveraged to create segmentation between networks, which allows for overlapping IP addressing to be used even on a single interface (that is, using subinterfaces), and because the traffic paths are isolated, network security is increased and can eliminate the need for encryption and authentication for network traffic.

Service Providers with Multiprotocol Label Switching (MPLS) backbones typically use VRFs to create separate virtual private networks (VPNs) for their customers, and when used in this manner, VRFs are known as *VPN Routing and Forwarding*.

When VRF is not used in conjunction with MPLS, it is known as *VRF-Lite* (also termed *multi-VRF CE*, or *multi-VRF customer-edge device*). Because MPLS is beyond the scope of this book, only VRF-Lite is covered in this section and is referred to it simply as VRF.

The configurations in Example 3-2 should help clarify the VRF concept. Example 3-2 shows how configuring different interfaces with overlapping IP addresses and subnets is not allowed within a routing table, not even if they are both on different interfaces because they would end up in the same routing table and cause a conflict.

Example 3-2 *Overlapping IP Address Problems*

```
IOS
R1(config)#interface GigabitEthernet0/1
R1(config-if)#ip address 10.0.3.1 255.255.255.0
R1(config-if)#interface GigabitEthernet0/3
R1(config-if)#ip address 10.0.3.2 255.255.255.0
```

```
% 10.0.3.0 overlaps with GigabitEthernet0/1
```

```
IOS XR
RP/0/0/CPU0:XR2(config)#interface gigabitEthernet 0/0/0/5
RP/0/0/CPU0:XR2(config-if)#ipv4 address 10.0.3.1/24
RP/0/0/CPU0:XR2(config-if)#commit
RP/0/0/CPU0:XR2(config-if)#
RP/0/0/CPU0:XR2(config)#interface gigabitEthernet 0/0/0/3
RP/0/0/CPU0:XR2(config-if)#ipv4 address 10.0.3.2/24
RP/0/0/CPU0:XR2(config-if)#commit

RP/0/0/CPU0:Jan 13 18:55:35.643 : ipv4_arm[189]: %IP-IP_ARM-3-CFLCT_FORCED_DOWN :
The IPv4 address 10.0.3.1/24 on GigabitEthernet0/0/0/5 conflicts with other IPv4
addresses and has been forced down
```

Note In IOS XR, the IP Address Repository Manager (IPARM) enforces the uniqueness of global IP addresses configured in the system. By default, when there is an IP address and subnet mask conflict, the lowest rack/slot/interface (that is, g0/0/0/3 is lower than g0/0/0/5) is the one that gets assigned the IP address. To change the default behavior, use the **ipv4 conflict-policy** {**static | highest-ip | longest-prefix**} command.

In older IOS releases, only single-protocol IPv4-only VRFs could be created. The command **ip vrf** *vrf-name* created a single-protocol VRF on the router and was activated on an interface with the command **ip vrf forwarding** *vrf-name* under the interface configuration mode.

In current IOS releases, a new configuration option allows the creation of multiprotocol VRFs that support both IPv4 and IPv6. Entering the command **vrf definition** *vrf-name* creates the multiprotocol VRF. Under VRF definition submode, the command **address-family** {**ipv4 | ipv6**} is required to specify the appropriate address family. The VRF is then associated to the interface with the command **vrf forwarding** *vrf-name* under the interface configuration submode.

Note The commands **ip vrf** *vrf-name* and **ip vrf forwarding** *vrf-name* will be available for a period of time before they are deprecated. To migrate any older IPv4-only VRFs to the new multiprotocol VRF configuration, you can use the **vrf upgrade-cli multi-af-mode** {**common-policies | non-common-policies**} [**vrf** *vrf-name*] command. When creating a new VRF, even if it is just an IPv4-only VRF, Cisco recommends using the multiprotocol VRF **vrf definition** and **vrf forwarding** commands.

In IOS, the following steps are required to create a VRF and assign it to an interface:

Step 1. **Create a multiprotocol VRF.**

The multiprotocol VRF routing table is created with the command **vrf definition** *vrf-name*.

Step 2. **Identify the address family.**

Initialize the appropriate address family with the command **address-family** {**ipv4** | **ipv6**}. The address family can be IPv4, IPv6, or both.

Step 3. **Specify the interface to be associated with the VRF.**

Enter interface configuration submode and specify the interface to be associated with the VRF with the command **interface** *interface-type interface-number*.

Step 4. **Associate the VRF to the interface.**

The VRF is associated to the interface or subinterface by entering the command **vrf forwarding** *vrf-name* under interface configuration submode.

Step 5. **Configure an IP address on the interface or subinterface.**

The IP address can be IPv4, IPv6, or both. It is configured by entering the following commands:

```
IPv4
ip address ip-address subnet-mask [secondary]

IPv6
ipv6 address {ipv6-address/prefix-length | prefix-name sub-bits/
prefix-length}
```

Note On IOS nodes, the VRF needs to be associated to the interface first before configuring an IP address. If an IP address is already configured, and the VRF is associated to the interface, IOS will remove the IP address.

IOS XR supports only multiprotocol VRFs. The following steps are required to create a multiprotocol VRF and assign it to an interface on an IOS XR node:

Step 1. **Create a multiprotocol VRF.**

The multiprotocol VRF routing table is created with the command **vrf** *vrf-name*. The VRF name is arbitrary.

Step 2. **Identify the address family.**

Initialize the appropriate address family with the command **address-family** {**ipv4** | **ipv6**} **unicast**. The address family can be IPv4, IPv6, or both.

Step 3. **Specify the interface to be associated with the VRF.**

Enter interface configuration submode and specify the interface to be associated with the VRF with the command **interface** *interface-type interface-number*.

Step 4. **Associate the VRF with an interface or subinterface.**

The VRF is associated with the interface or subinterface by entering the command **vrf** *vrf-name* under interface configuration submode.

Step 5. **Configure an IP address on the interface or subinterface.**

The IP address can be IPv4, IPv6, or both. It is configured by entering the following commands:

```
IPv4
ipv4 address ipv4-address subnet-mask

IPv6
ipv6 address ipv6-address/prefix-length
```

Note For IOS XR, the VRF needs to be associated to the interface first before configuring an IP address; otherwise, the VRF configuration will not be accepted.

Figure 3-6 Illustrates two routers to help visualize the VRF routing table concept. One of the routers has no VRFs configured, and the other one has a management VRF named MGMT. This figure can be used as a reference for the following examples.

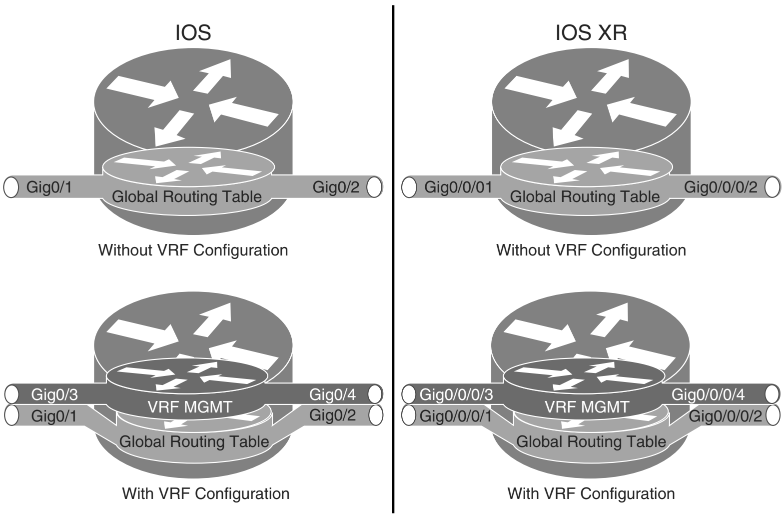


Figure 3-6 *Comparison of a Router with no VRFs and a Router with a VRF*

Table 3-3 provides a set of interfaces and IP addresses that overlap between the global routing table and the VRF. This information is used in the following examples.

Table 3-3 *Sample Interfaces and IP Addresses*

IOS Interface	IOS XR Interface	IP Address	VRF	Global
Gigabit Ethernet 0/1	Gigabit Ethernet 0/0/0/1	10.0.3.1/24		✓
Gigabit Ethernet 0/2	Gigabit Ethernet 0/0/0/2	10.0.4.1/24		✓
Gigabit Ethernet 0/3	Gigabit Ethernet 0/0/0/3	10.0.3.1/24	MGMT	
Gigabit Ethernet 0/4	Gigabit Ethernet 0/0/0/4	10.0.4.1/24	MGMT	

Example 3-3 shows how the IP addresses are assigned to the interfaces in the global routing table shown in Table 3-3.

Example 3-3 *IP Address Configuration in Global Routing Table*

```
IOS
R1(config)#interface GigabitEthernet0/1
R1(config-if)#ip address 10.0.3.1 255.255.255.0
R1(config)#interface GigabitEthernet0/2
R1(config-if)#ip address 10.0.4.1 255.255.255.0
```

```
IOS XR
RP/0/0/CPU0:XR1(config)#interface gigabitEthernet 0/0/0/1
RP/0/0/CPU0:XR1(config-if)#ipv4 address 10.0.3.1/24
RP/0/0/CPU0:XR1(config)#interface gigabitEthernet 0/0/0/2
RP/0/0/CPU0:XR1(config-if)#ipv4 address 10.0.4.1/24
RP/0/0/CPU0:XR1(config-if)#commit
```

Example 3-4 displays the global routing table with the command **show ip route** for IOS and **show route** for IOS XR to show the IP addresses configured in Example 3-3.

Example 3-4 *Output of Global Routing Table*

```
IOS
R1#show ip route
! Output omitted for brevity
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.3.0/24 is directly connected, GigabitEthernet0/1
L        10.0.3.1/32 is directly connected, GigabitEthernet0/1
C        10.0.4.0/24 is directly connected, GigabitEthernet0/2
L        10.0.4.1/32 is directly connected, GigabitEthernet0/2
```

```
IOS XR
RP/0/0/CPU0:XR1#show route
```

```
! Output omitted for brevity

C    10.0.3.0/24 is directly connected, 00:00:25, GigabitEthernet0/0/0/1
L    10.0.3.1/32 is directly connected, 00:00:25, GigabitEthernet0/0/0/1
C    10.0.4.0/24 is directly connected, 00:00:02, GigabitEthernet0/0/0/2
L    10.0.4.1/32 is directly connected, 00:00:02, GigabitEthernet0/0/0/2
```

Example 3-5 shows how the VRF named MGMT is created, two interfaces are associated with it, and the IP addresses in Table 3-3 are configured on the interfaces. These IP addresses overlap with the ones configured in Example 3-3, but there is no conflict because they are in a different routing table.

Example 3-5 *VRF Configuration Example*

```
IOS
R1(config)#vrf definition MGMT
R1(config-vrf)# address-family ipv4
R1(config)#interface GigabitEthernet0/3
R1(config-if)#vrf forwarding MGMT
R1(config-if)#ip address 10.0.3.1 255.255.255.0
R1(config)#interface GigabitEthernet0/4
R1(config-if)#vrf forwarding MGMT
R1(config-if)#ip address 10.0.4.1 255.255.255.0
```

```
IOS XR
RP/0/0/CPU0:XR1(config)#vrf MGMT address-family ipv4 unicast
RP/0/0/CPU0:XR1(config-vrf-af)#root
RP/0/0/CPU0:XR1(config)#interface gigabitEthernet 0/0/0/3
RP/0/0/CPU0:XR1(config-if)#vrf MGMT
RP/0/0/CPU0:XR1(config-if)#ipv4 address 10.0.3.1/24
RP/0/0/CPU0:XR1(config)#interface gigabitEthernet 0/0/0/4
RP/0/0/CPU0:XR1(config-if)#vrf MGMT
RP/0/0/CPU0:XR1(config-if)#ipv4 address 10.0.4.1/24
RP/0/0/CPU0:XR1(config-if)#commit
```

Example 3-6 shows how the VRF IP addresses configured in Example 3-5 cannot be seen in the output of the **show ip route** command for IOS and the **show route** command for IOS XR; these commands display only the contents of the global routing table. To see a VRF routing table, the commands **show ip route vrf** *vrf-name* for IOS and **show route vrf** {all | *vrf-name*} for IOS XR should be used.

Example 3-6 *Output of Global Routing Table and VRF Routing Table*

```
R1#show ip route
! Output omitted for brevity
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.3.0/24 is directly connected, GigabitEthernet0/1
L        10.0.3.1/32 is directly connected, GigabitEthernet0/1
C        10.0.4.0/24 is directly connected, GigabitEthernet0/2
L        10.0.4.1/32 is directly connected, GigabitEthernet0/2

R1#show ip route vrf MGMT
! Output omitted for brevity
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.3.0/24 is directly connected, GigabitEthernet0/3
L        10.0.3.1/32 is directly connected, GigabitEthernet0/3
C        10.0.4.0/24 is directly connected, GigabitEthernet0/4
L        10.0.4.1/32 is directly connected, GigabitEthernet0/4
```

```
RP/0/0/CPU0:XR1#show route
! Output omitted for brevity
C    10.0.3.0/24 is directly connected, 00:12:44, GigabitEthernet0/0/0/1
L    10.0.3.1/32 is directly connected, 00:12:44, GigabitEthernet0/0/0/1
C    10.0.4.0/24 is directly connected, 00:12:21, GigabitEthernet0/0/0/2
L    10.0.4.1/32 is directly connected, 00:12:21, GigabitEthernet0/0/0/2

RP/0/0/CPU0:XR1#show route vrf MGMT
! Output omitted for brevity
C    10.0.3.0/24 is directly connected, 00:09:15, GigabitEthernet0/0/0/3
L    10.0.3.1/32 is directly connected, 00:09:15, GigabitEthernet0/0/0/3
C    10.0.4.0/24 is directly connected, 00:00:10, GigabitEthernet0/0/0/4
L    10.0.4.1/32 is directly connected, 00:00:10, GigabitEthernet0/0/0/4
```

In IOS, to display a quick summary of the usability status for each IP interface, in addition to all the IP addresses configured in the global routing table and all VRFs, the command **show ip interface brief** should be used. In IOS XR, the command **show ipv4 interface brief** only shows the IP addresses in the global routing table. To see the IP addresses in the global routing table and all VRFs, use the command **show ipv4 vrf all interface brief.** Example 3-7 provides sample output of these **show** commands.

Example 3-7 *Verification of Interfaces Status and IP Addresses*

```
R1#show ip interface brief

Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/1     10.0.3.1        YES NVRAM  up                    up
```

```
GigabitEthernet0/2      10.0.4.1        YES NVRAM  up                    up
GigabitEthernet0/3      10.0.3.1        YES NVRAM  up                    up
GigabitEthernet0/4      10.0.4.1        YES NVRAM  up                    up
```

```
RP/0/0/CPU0:XR2#show ipv4 interface brief

Interface                      IP-Address      Status                Protocol
GigabitEthernet0/0/0/0         unassigned      Shutdown              Down
GigabitEthernet0/0/0/1         10.0.3.1        Up                    Up
GigabitEthernet0/0/0/2         10.0.4.1        Up                    Up

RP/0/0/CPU0:XR2#show ipv4 vrf all interface brief

Interface                      IP-Address      Status          Protocol Vrf-Name
GigabitEthernet0/0/0/0       unassigned    Shutdown     Down      default
GigabitEthernet0/0/0/1       10.0.3.1      Up           Up        default
GigabitEthernet0/0/0/2       10.0.4.1      Up           Up        default
GigabitEthernet0/0/0/3       10.0.3.1      Up           Up        MGMT
GigabitEthernet0/0/0/4       10.0.4.1      Up           Up        MGMT
```

VRF-Lite can provide similar functionality to that of virtual local-area networks (VLANs); however, instead of relying on Layer 2 technologies such as spanning tree, Layer 3 dynamic routing protocols can be used. Using routing protocols over Layer 2 technologies has some advantages such as improved network convergence times, dynamic traffic load sharing, and troubleshooting tools such as ping and traceroute.

IP Packet Switching

Chapter 2, "IP Addressing," explained that devices on the same subnet could communicate directly with each other without the need of a router. The second layer of the OSI model, the data link layer, handles addressing beneath the IP protocol stack so that communication is directed between hosts. Network packets include the Layer 2 addressing with unique source and destination addresses for that segment. Ethernet commonly uses MAC addresses, and other data link layer protocols such as Frame Relay use an entirely different method of Layer 2 addressing.

The first routers would receive a packet, remove the Layer 2 information, and verify that the route exists for the destination IP address. If a matching route could not be found, the packet was dropped. If a matching route was found, the router would identify it and add new Layer 2 information to the packet. The Layer 2 source address would be the router's outbound interface, and the destination information would be next hop's Layer 2 address.

Figure 3-7 illustrates the concept where PC A is sending a packet to PC B via Ethernet connection to R1. PC A sends the packet to R1's MAC address of 00:C1:5C: 00:00:02. R1 receives the packet, removes the Layer 2 information, and looks for a route to the

192.168.2.2 address. R1 identifies that connectivity to the 192.168.2.2 IP address is through Gigabit Ethernet 0/1. R1 adds the Layer 2 source address using its Gigabit Ethernet 0/1's MAC address 00:C1:5C:00:00:03 and a destination address for PC B of 00:00:00:00:00:04.

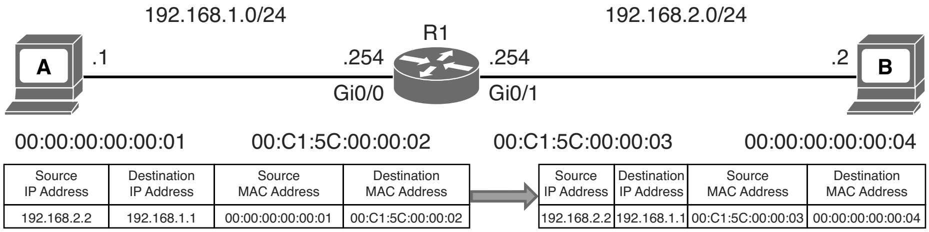


Figure 3-7 *Layer 2 Addressing*

Advancement in technologies has streamlined the process so that routers do not remove and add the Layer 2 addressing but simply rewrites them. IP packet switching or IP packet forwarding is the faster process of receiving an IP packet on an input interface and making a decision of whether to forward the packet to an output interface or drop it. This process is simple and streamlined for a router to be able to forward large amounts of packets.

When the first Cisco routers were developed, they used a mechanism called *process switching* to switch the packets through the routers. As network devices evolved, Cisco created Fast Switching and Cisco Express Forwarding (CEF) to optimize the switching process for the routers to be able to handle larger packet volumes. Fast Switching is deprecated in newer IOS releases and is not covered in this book.

Process Switching

Process switching, also referred to as *software switching* or *slow path*, is the switching mechanism in which the general-purpose CPU on a router is in charge of packet switching. In IOS, the ip_input process runs on the general-purpose CPU for processing incoming IP packets. Process switching is the fallback for CEF because it is dedicated for processing punted IP packets when they cannot be switched by CEF.

In IOS XR, the Network Input/Output (NetIO) process is the equivalent to the IOS ip_input process and is responsible for forwarding packets in software.

The type of packets that require software handling for both IOS and IOS XR include the following:

- Packets sourced or destined to the router (that is, control traffic, routing protocols)
- Packets that are too complex for the hardware to handle (that is, IP packets with IP options)
- Packets that require extra information that is not currently known (that is, Address Resolution Protocol [ARP] resolution, and so on)

Note Software switching is significantly slower than switching done in hardware. NetIO is designed to handle a very small percentage of traffic handled by the system. Packets are hardware switched whenever possible.

Figure 3-8 illustrates how a packet that cannot be CEF switched is punted to the CPU for processing. The ip_input process consults the routing table and ARP table to obtain the next-hop router's IP address, outgoing interface, and MAC address. It then overwrites the destination MAC address of the packet with the next-hop router's MAC address, overwrites the source MAC address with the MAC address of the outgoing Layer 3 interface, decrements the IP Time-To-Live (TTL) field, recomputes the IP header checksum, and finally delivers the packet to the next-hop router.

The routing table, also known as the *Routing Information Base* (RIB), is built from information obtained from dynamic routing protocols, directly connected and static routes. The ARP table is built from information obtained from the ARP protocol. The ARP protocol is used by IP hosts to dynamically learn the MAC address of other IP hosts on the same subnet. For example, an IP host that needs to perform address resolution for another IP host connected by Ethernet can send an ARP request using a LAN broadcast address, and it then waits for an ARP reply from the IP host. The ARP reply includes the required Layer 2 physical MAC address information.

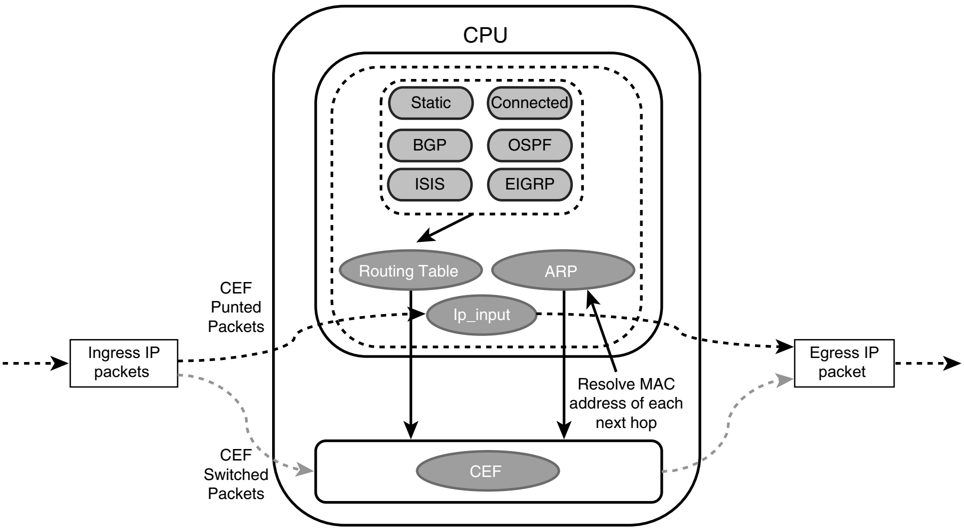


Figure 3-8 *Process Switching*

Cisco Express Forwarding

Cisco Express Forwarding (CEF) is a Cisco proprietary switching mechanism developed to keep up with the demands of evolving network infrastructures. It has been the default switching mechanism on most Cisco platforms that do all their packet switching using the general-purpose CPU (software based routers) since the 1990s, and it is the default switching mechanism used by all Cisco platforms that use specialized application specific integrated circuits (ASICs) and network processing units (NPUs) for high packet throughput (hardware-based routers).

The general-purpose CPU on the software-based and hardware-based routers is similar and perform all the same functions, the difference being that on software based routers the general-purpose CPU is in charge of all operations, including CEF switching (software CEF), and the hardware-based routers do CEF switching using *forwarding engines* that are implemented in specialized ASICs, TCAMs, and NPUs (hardware CEF). Forwarding engines provide the packet switching, forwarding, and route lookup capability to routers.

Given the low cost of the general-purpose CPUs, the price point of software-based routers will be much more affordable, but at the expense of total packet throughput.

When a route processor (RP) engine is equipped with a forwarding engine so that it can make all the packet switching decisions, this is known as a *centralized forwarding architecture*. If the line cards are equipped with forwarding engines so that they can make packet switching decision without intervention of the RP, this is known as a *distributed forwarding architecture*.

For a centralized forwarding architecture, when a packet is received on the ingress line card, it is transmitted to the forwarding engine on the RP. The forwarding engine examines the packet's headers and determines that the packet will be sent out a port on the egress line card, and forwards the packet to the egress line card to be forwarded.

For a distributed forwarding architecture, when a packet is received on the ingress line card, it is transmitted to the local forwarding engine. The forwarding engine performs a packet lookup, and if it determines that the outbound interface is local, it forwards the packet out a local interface. If the outbound interface is located on a different line card, the packet is sent across the switch fabric, also known as the *backplane*, directly to the egress line card, bypassing the RP.

Figure 3-9 illustrates a packet flowing across a centralized and a distributed forwarding architecture.

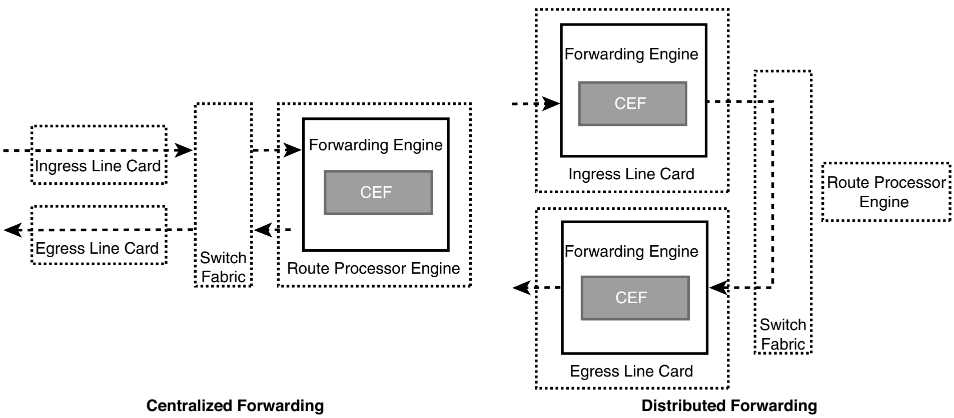


Figure 3-9 *Centralized Versus Distributed Forwarding Architectures*

Software CEF

Software CEF, also known as the software *Forwarding Information Base* (FIB), consists of the following components:

- **Forwarding Information Base:** The FIB is built directly from the routing table and contains the next-hop IP address for each destination IP in the network. It keeps a mirror image of the forwarding information contained in the IP routing table. When a routing or topology change occurs in the network, the IP routing table is updated, and these changes are reflected in the FIB. CEF uses the FIB to make IP destination prefix-based switching decisions
- **Adjacency table:** The adjacency table is also known as the *Adjacency Information Base (*AIB). It contains the MAC addresses and egress interfaces of all directly connected next hops, and it is populated with data from the ARP table and other Layer 2 protocol tables (that is, Frame Relay map tables).

Figure 3-10 illustrates how the CEF table is built from the routing table and the ARP table and how a packet is CEF switched through the router. When an IP packet is received, if there is a valid FIB and adjacency table entry for it, the router overwrites the destination MAC address of the packet with the next hop router's MAC address, overwrites the source MAC address with the MAC address of the outgoing Layer 3 interface, decrements IP TTL field, recomputes the IP header checksum, and finally delivers the packet to the next-hop router.

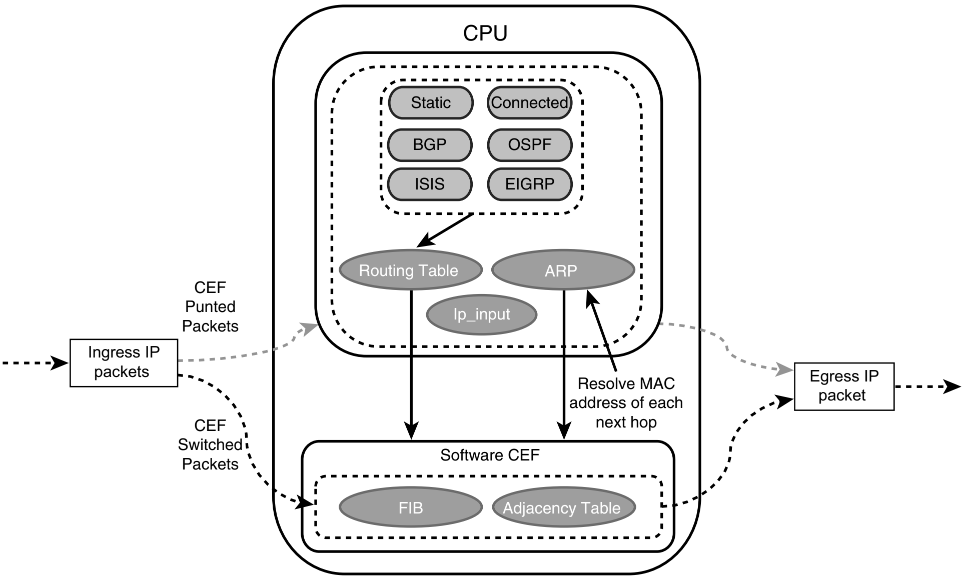


Figure 3-10 *CEF Switching*

Hardware CEF

The ASICs in hardware-based routers have a very high cost to design, produce, and troubleshoot. ASICs allow for very high packet rates, but the trade-off is that they are limited in their functionality because they are hardwired to perform specific tasks. There are routers equipped with NPUs that are designed to overcome the inflexibility of ASICs. Unlike ASICs, NPUs are programmable, and their firmware can be changed with relative ease.

The main advantage of the distributed forwarding architectures is that the packet throughput performance is greatly improved by offloading the packet switching responsibilities to the line cards. Packet switching in distributed architecture platforms is done via distributed CEF (dCEF), which is a mechanism in which the CEF data structures are downloaded to forwarding ASICs and the CPUs of all line cards so that they can participate in packet switching; this allows for the switching to be done at the distributed level, thus increasing the packet throughput of the router.

Software CEF in hardware-based platforms is not used to do packet switching as in software-based platforms; instead, it is used to program the hardware CEF, as shown in Figure 3-11.

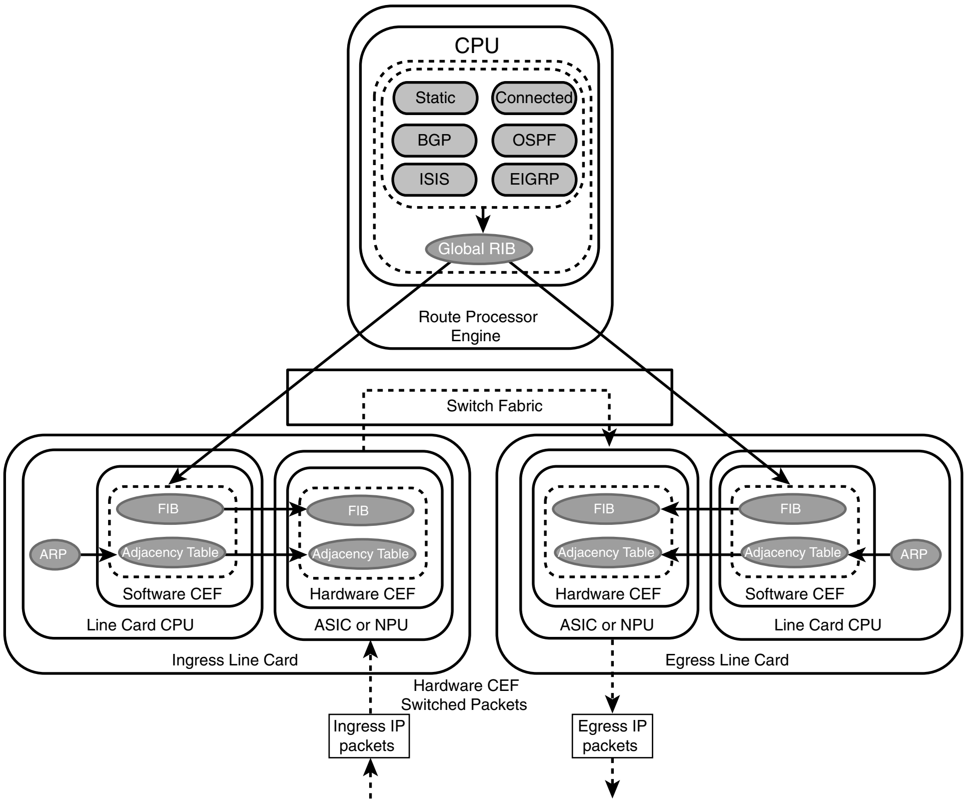


Figure 3-11 *dCEF Hardware Switching*

Figure 3-11 also illustrates how the RIB process interacts with the RIBs of the routing protocols. The RIB process is in charge of the calculation of best paths, alternative paths, and the redistribution from different protocols and all these details merge into the global RIB (gRIB), where the best path for a destination network is installed. This is further distributed into the software CEF tables of different line cards, which is further mirrored into hardware CEF. The Switch Fabric is the backplane for all modules in the system. It creates a dedicated connection between all line cards and the route processors and provides fast data switching transmission between them.

In most distributed architecture platforms, if the incoming packet is control plane traffic or management traffic it is punted to the RP's CPU. The following list includes some examples of packets that are typically punted for processing by the RP's CPU or line card's CPU:

- Control traffic, such as BGP, OSPF, IS-IS, PIM, IGMP, and so on
- Management traffic, such as Telnet, SSH, SNMP, and so on
- Layer 2 mechanisms, such as CDP, ARP, LACP PDU, BFD, and so on
- Fragmentation, DF bit set, IP options set
- TTL expired
- ICMP echo request

Planes of Operation

A router is typically segmented into three planes of operation, each with a specific and clearly defined objective:

- **The control plane:** The control plane is the brain of the router. It consists of dynamic IP routing protocols (that is OSPF, IS-IS, BGP, and so on), the RIB, routing updates, in addition to other protocols such as PIM, IGMP, ICMP, ARP, BFD, LACP, and so on. In short, the control plane is responsible for maintaining sessions and exchanging protocol information with other router or network devices.

In centralized architecture platforms, the general-purpose CPU manages all control plane protocols. In distributed architecture platforms, routing protocols, and most other protocols, always run on the core CPU in the RPs or Supervisor engines, but there are other control plane protocols such as ARP, BFD, and ICMP that in some distributed architecture platforms have now been offloaded to the line card CPU.

- **The data plane:** The data plane is the forwarding plane, which is responsible for the switching of packets through the router (that is, process switching and CEF switching). In the data plane, there could be features that could affect packet forwarding such as quality of service (QoS) and access control lists (ACLs).
- **The management plane:** The management plane is used to manage a device through its connection to the network. Examples of protocols processed in the management plane include Simple Network Management Protocol (SNMP), Telnet, File Transfer Protocol (FTP), Secure FTP, and Secure Shell (SSH). These management protocols are used for monitoring and for command-line interface (CLI) access.

Figure 3-12 shows how the three planes of operation and how the processes are isolated from each other. In IOS XR, a process failure within one plane does not affect other processes or applications within that plane. This layered architecture creates a more reliable model than one with a monolithic architecture such as IOS, where failure of a single process may cause a failure of the whole system.

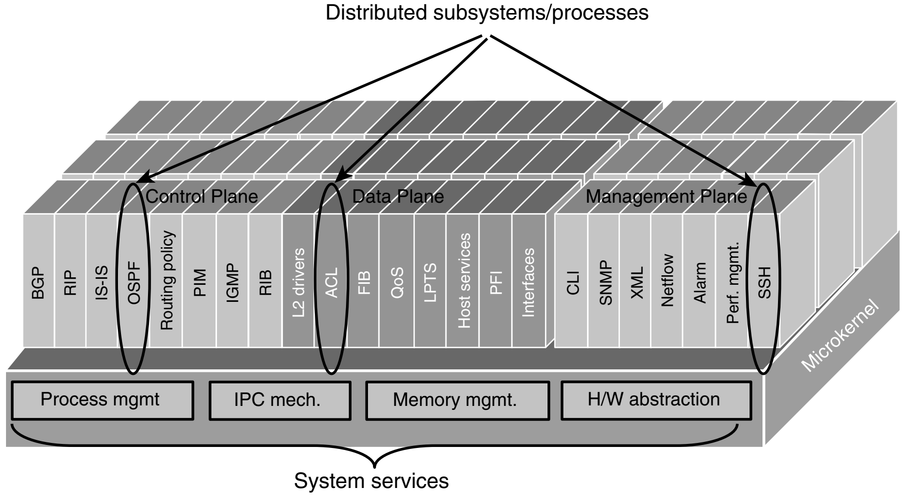


Figure 3-12 *Separation of Control, Data, and Management Planes*

Summary

This chapter provided an overview of the fundamentals of IP routing and IP switching and the control planes of operation. In summary, it showed how a router makes a forwarding decision, which consists of three basic components:

- The routing protocols, which are used to build the routing table (RIB)
- The routing table, which is used to program the switching mechanisms (that is, CEF)
- The switching mechanisms used to perform the actual packet forwarding

References in This Chapter

Bollapragada, Vijay, Russ White, and Curtis Murphy. *Inside Cisco IOS Software Architecture*. (ISBN-13: 978-1587058165).

Stringfield, Nakia, Russ White, and Stacia McKee. *Cisco Express Forwarding*. (ISBN-13: 978-0-13-343334-0).

Tahir, Mobeen, Mark Ghattas, Dawit Birhanu, and Syed Natif Nawaz. *Cisco IOS XR Fundamentals*. (ISBN-13: 978-1-58705-271-2).

Doyle, Jeff and Jennifer Carroll. *Routing TCP/IP, Volume 1*, 2nd Ed. (ISBN-13: 978-1-58705-202-6).

This page intentionally left blank

Index

Symbols

A

B

C

D

E

F

G

H

I

J - K

L

M

N

O

P

Q

R

S

T

U

V

W