Troubleshooting BGP
A Practical Guide to Understanding and Troubleshooting BGP

Vinit Jain, CCIE No. 22854
Brad Edgeworth, CCIE No. 31574
Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.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 email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Editor-in-Chief: Mark Taub
Alliances Manager, Cisco Press: Ron Fligge
Product Line Manager: Brett Bartow
Managing Editor: Sandra Schroeder
Development Editor: Marianne Bartow
Senior Project Editor: Tonya Simpson
Copy Editor: Barbara Hacha

Technical Editors: Richard Furr, Ramiro Garza Rios
Editorial Assistant: Vanessa Evans
Cover Designer: Chuti Prasertsith
Composition: codeMantra
Indexer: Cheryl Lenser
Proofreader: Deepa Ramesh
About the Authors

Vinit Jain, CCIE No. 22854 (R&S, SP, Security & DC), is a High Touch Technical Support (HTTS) engineer with Cisco providing support to premium customers of Cisco on complex routing technologies. Before joining Cisco, Vinit worked as a CCIE trainer and a network consultant. In addition to his expertise in networks, he has experience with software development, with which he began his career.

Vinit holds certifications for multiple vendors, such as Cisco, Microsoft, Sun Microsystems, VMware, and Oracle, and also is a Certified Ethical Hacker. Vinit is a speaker at Cisco Live and various other forums, including NANOG. Vinit pursued his graduation from Delhi University in Mathematics and earned his Masters in Information Technology from Kuvempu University in India. Vinit is married and is presently based out of RTP, North Carolina. Vinit can be found on Twitter @vinugenie.

Brad Edgeworth, CCIE No. 31574 (R&S & SP), has been with Cisco working as a systems engineer and a technical leader. Brad is a distinguished speaker at Cisco Live, where he has presented on multiple topics. Before joining Cisco, Brad worked as a network architect and consulted 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 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. Brad can be found on Twitter @BradEdgeworth.
About the Technical Reviewers

Richard Furr, CCIE No. 9173 (R&S & SP), is a technical leader with the Cisco Technical Assistance Center (TAC). For the past 15 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.

Ramiro Garza Rios, CCIE No. 15469 (R&S, SP, and Security), is a solutions integration architect with Cisco Advanced Services, where he plans, designs, implements, and optimizes IP NGN service provider networks. Before joining Cisco in 2005, he was a network consulting and presales engineer for a Cisco Gold Partner in Mexico, where he planned, designed, and implemented both enterprise and service provider networks.
Dedications

I would like to dedicate this book to my brother, Lalit, who is the inspiration and driving force behind everything I have achieved.

—Vinit

This book is dedicated to my family. Thank you both for letting me sleep in after a late-night writing session. To my wife, Tanya, “The Queen of Catan,” thank you for bringing joy to my life. To my daughter, Teagan, listen to your mother. She is almost always right, and way better with her grammar than I am.

—Brad

Acknowledgments

Vinit Jain:

I would like to thank Russ White, Carlos Pignataro, Richard Furr, Pete Lumbis, Alejandro Eguiarte, and Brett Bartow for making this book possible.

I’d like to give special recognition to Alvaro Retana, Xander Thujis, and Steven Cheung for providing expert technical knowledge and advice on various topics, making this book more useful and close to real-life troubleshooting scenarios.

To our technical editors, Richard and Ramiro. In addition to your technical accuracy, your insight into the technologies needed versus and different perspective has kept the size of the book manageable.

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 to my managers, Ruwani Biggers and Chip Little, who have helped me with this adventurous and fun-filled project.

Brad Edgeworth:

A debt of gratitude goes toward my co-author, Vinit. Thank you for allowing me to work on this book with you, although we spent way too many nights on the phone at 1 a.m. Your knowledge and input made this a better book.

To our technical editors, Richard and Ramiro. Thank you for finding all of our mistakes. Not that we had many, but you still saved us a couple times. I won’t tell if you won’t.

A special thank you goes to Brett Bartow and the Cisco Press team. You are the “magicians” that make this book look as good as it does!

A special thanks goes to Craig Smith. “You are so money, and you don’t even know it!” To my co-workers Rob, John, and Gregg. Yes, this means I probably will need to go on another “book signing tour.” If anything breaks while I’m gone, order a queso and chips!
# Contents at a Glance

Foreword xxii
Introduction xxiii

## Part I BGP Fundamentals
Chapter 1 BGP Fundamentals 1

## Part II Common BGP Troubleshooting
Chapter 2 Generic Troubleshooting Methodologies 47
Chapter 3 Troubleshooting Peering Issues 83
Chapter 4 Troubleshooting Route Advertisement and BGP Policies 145
Chapter 5 Troubleshooting BGP Convergence 205

## Part III BGP Scalability Issues
Chapter 6 Troubleshooting Platform Issues Due to BGP 251
Chapter 7 Scaling BGP 283
Chapter 8 Troubleshooting BGP Edge Architectures 367

## Part IV Securing BGP
Chapter 9 Securing BGP 419

## Part V Multiprotocol BGP
Chapter 10 MPLS Layer 3 VPN (L3VPN) 481
Chapter 11 BGP for MPLS L2VPN Services 543
Chapter 12 IPv6 BGP for Service Providers 591
Chapter 13 VxLAN BGP EVPN 641

## Part VI High Availability
Chapter 14 BGP High Availability 693

## Part VII BGP: Looking Forward
Chapter 15 Enhancements in BGP 755
Index 789
Contents

Foreword xxii
Introduction xxiii

Part I  BGP Fundamentals

Chapter 1  BGP Fundamentals  1
Border Gateway Protocol  1
  Autonomous System Numbers  2
  Path Attributes  3
  Loop Prevention  3
  Address Families  3
  BGP Sessions  4
Inter-Router Communication  5
BGP Messages  6
  OPEN  6
  Hold Time  6
  BGP Identifier  7
  KEEPALIVE  7
  UPDATE  7
  NOTIFICATION Message  8
BGP Neighbor States  8
  Idle  9
  Connect  9
  Active  10
  OpenSent  10
  OpenConfirm  10
  Established  10
Basic BGP Configuration  11
  IOS  11
  IOS XR  12
  NX-OS  13
  Verification of BGP Sessions  14
  Prefix Advertisement  17
  BGP Best-Path Calculation  20
Route Filtering and Manipulation  21
Chapter 3  Troubleshooting Peering Issues  83

BGP Peering Down Issues  83
  Verifying Configuration  84
  Verifying Reachability  87
  *Find the Location and Direction of Packet Loss*  88
  *Verify Whether Packets Are Being Transmitted*  89
  *Use Access Control Lists to Verify Whether Packets Are Received*  90
  *Check ACLs and Firewalls in Path*  91
  *Verify TCP Sessions*  94
  *Simulate a BGP Session*  95
  Demystifying BGP Notifications  96
  Decode BGP Messages  99
  Troubleshoot Blocked Process in IOS XR  103
  *Verify BGP and BPM Process State*  104
  *Verify Blocked Processes*  105
  *Restarting a Process*  106
  BGP Traces in IOS XR  106
  BGP Traces in NX-OS  108
  Debugs for BGP  110
  Troubleshooting IPv6 Peers  112
  Case Study—Single Session Versus Multisession  113
  *Multisession Capability*  114
  *Single-Session Capability*  115
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Peer Flapping Issues</td>
<td>115</td>
</tr>
<tr>
<td>Bad BGP Update</td>
<td>115</td>
</tr>
<tr>
<td>Hold Timer Expired</td>
<td>116</td>
</tr>
<tr>
<td>Interface Issues</td>
<td>116</td>
</tr>
<tr>
<td>Physical Connectivity</td>
<td>117</td>
</tr>
<tr>
<td>Physical Interface</td>
<td>117</td>
</tr>
<tr>
<td>Input Hold Queue</td>
<td>117</td>
</tr>
<tr>
<td>TCP Receive Queue</td>
<td>119</td>
</tr>
<tr>
<td>MTU Mismatch Issues</td>
<td>120</td>
</tr>
<tr>
<td>High CPU Causing Control-Plane Flaps</td>
<td>125</td>
</tr>
<tr>
<td>Control Plane Policing</td>
<td>127</td>
</tr>
<tr>
<td>CoPP on NX-OS</td>
<td>129</td>
</tr>
<tr>
<td>Local Packet Transport Services</td>
<td>134</td>
</tr>
<tr>
<td>Dynamic BGP Peering</td>
<td>138</td>
</tr>
<tr>
<td>Dynamic BGP Peer Configuration</td>
<td>139</td>
</tr>
<tr>
<td>Dynamic BGP Challenges</td>
<td>142</td>
</tr>
<tr>
<td>Misconfigured MD5 Password</td>
<td>142</td>
</tr>
<tr>
<td>Resource Issues in a Scaled Environment</td>
<td>142</td>
</tr>
<tr>
<td>TCP Starvation</td>
<td>142</td>
</tr>
<tr>
<td>Summary</td>
<td>143</td>
</tr>
<tr>
<td>References</td>
<td>143</td>
</tr>
</tbody>
</table>

**Chapter 4**  **Troubleshooting Route Advertisement and BGP Policies**  **145**

Troubleshooting BGP Route Advertisement  145
- Local Route Advertisement Issues  145
- Route Aggregation Issues  147
- Route Redistribution Issues  150
- BGP Tables  152
- Receiving and Viewing Routes  154

Troubleshooting Missing BGP Routes  156
- Next-Hop Check Failures  157
- Bad Network Design  160
- Validity Check Failure  162
- AS-Path  162
- Originator-ID/Cluster-ID  165
- BGP Communities  167
- BGP Communities: No-Advertise  167
- BGP Communities: No-Export  169
Chapter 5  Troubleshooting BGP Convergence  205

Understanding BGP Route Convergence  205
  BGP Update Groups  207
  BGP Update Generation  212
Troubleshooting Convergence Issues  216
  Faster Detection of Failures  218
Jumbo MTU for Faster Convergence 219
Slow Convergence due to Periodic BGP Scan 219
Slow Convergence due to Default Route in RIB 222
BGP Next-Hop Tracking 223
Selective Next-Hop Tracking 225
Slow Convergence due to Advertisement Interval 226
Computing and Installing New Path 226
Troubleshooting BGP Convergence on IOS XR 227
Verifying Convergence During Initial Bring Up 227
Verifying BGP Reconvergence in Steady State Network 228
Troubleshooting BGP Convergence on NX-OS 234
BGP Slow Peer 237
BGP Slow Peer Symptoms 238
High CPU due to BGP Router Process 238
Traffic Black Hole and Missing Prefixes in BGP table 238
BGP Slow Peer Detection 239
Verifying OutQ value 240
Verifying SndWnd 240
Verifying Cache Size and Pending Replication Messages 241
Workaround 242
Changing Outbound Policy 242
Advertisement Interval 243
BGP Slow Peer Feature 245
Static Slow Peer 245
Dynamic Slow Peer Detection 245
Slow Peer Protection 246
Slow Peer Show Commands 246
Troubleshooting BGP Route Flapping 246
Summary 250
Reference 250

Part III BGP Scalability Issues

Chapter 6 Troubleshooting Platform Issues Due to BGP 251
Troubleshooting High CPU Utilization due to BGP 251
Troubleshooting High CPU due to BGP on Cisco IOS 252
High CPU due to BGP Scanner Process 253
High CPU due to BGP Router Process 255
High CPU Utilization due to BGP I/O Process 256
Troubleshooting High CPU due to BGP on IOS XR  258
Troubleshooting High CPU due to BGP on NX-OS  262
Capturing CPU History  265
Troubleshooting Sporadic High CPU Condition  265
Troubleshooting Memory Issues due to BGP  267
TCAM Memory  269
Troubleshooting Memory Issues on Cisco IOS Software  269
Troubleshooting Memory Issues on IOS XR  274
Troubleshooting Memory Issues on NX-OS  278
Restarting Process  281
Summary  281
References  282

Chapter 7  Scaling BGP  283
The Impact of Growing Internet Routing Tables  283
Scaling Internet Table on Various Cisco Platforms  285
Scaling BGP Functions  288
  Tuning BGP Memory  290
  Prefixes  290
  Managing the Internet Routing Table  290
  Paths  292
  Attributes  293
  Tuning BGP CPU  295
  IOS Peer-Groups  295
  IOS XR BGP Templates  295
  NX-OS BGP Peer Templates  296
  BGP Peer Templates on Cisco IOS  297
  Soft Reconfiguration Inbound Versus Route Refresh  298
  Dynamic Refresh Update Group  302
  Enhanced Route Refresh Capability  305
  Outbound Route Filtering (ORF)  309
  Prefix-Based ORF  309
  Extended Community-Based ORF  309
  BGP ORF Format  310
  BGP ORF Configuration Example  312
  Maximum Prefixes  316
  BGP Max AS  318
  BGP Maximum Neighbors  322
Scaling BGP with Route Reflectors  322
BGP Route Reflector Clusters  324
Hierarchical Route Reflectors  331
Partitioned Route Reflectors  332
BGP Selective Route Download  339
Virtual Route Reflectors  342
BGP Diverse Path  346
Shadow Route Reflectors  349
Shadow Sessions  355
Route Servers  357
Summary  364
References  365

Chapter 8  Troubleshooting BGP Edge Architectures  367
BGP Multihoming and Multipath  367
  Resiliency in Service Providers  370
  EBGP and IBGP Multipath Configuration  370
  EIBGP Multipath  372
R1  373
R2  374
R3  374
R4  375
R5  376
  AS-Path Relax  377
Understanding BGP Path Selection  377
  Routing Path Selection Longest Match  377
BGP Best-Path Overview  379
Weight  380
  Local Preference  380
  Locally Originated via Network or Aggregate Advertisement  380
  Accumulated Interior Gateway Protocol (AIGP)  381
Shortest AS-Path  383
Origin Type  383
  Multi-Exit Discriminator (MED)  384
EBGP over IBGP  386
  Lowest IGP Metric  386
  Prefer the Oldest EBGP Path  387
Router ID  387
Icons Used in This Book

Ethernet Circuit
Network
ASA Firewall
Layer 2 Switch
Multi-Layer Switch
Router
IOS XR
Nexus Device
Leaf Device
Spine Device
Workstation
Server
Security Server
DDOS Analyzer
Redistribution

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. 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).
- **Italic** indicates arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate an optional element.
- Braces ({ }) indicate a required choice.
- Braces within brackets ([|]) indicate a required choice within an optional element.
Foreword

The Internet has revolutionized the world by providing an unlimited supply of information to a user’s fingertips in a matter of seconds, or connecting people halfway around the world with voice and video calls. More people are using the Internet in ways unimaginable when it was first conceived. The size of the Internet routing prohibits the use of almost any routing protocol except for BGP.

More and more organizations continue to deploy BGP across every vertical, segment, and corner of the Earth because there have been so many new features and technologies introduced to BGP. BGP is not only used by the service providers but has become a fundamental technology in enterprises and data centers.

As the leader of Cisco’s technical services for more than 25 years, I have the benefit of working with the best network professionals in the industry. This book is written by Vinit and Brad, two “Network Rock Stars,” who have been in my organization for years supporting multiple Cisco customers. Vinit continues to provide dedicated service to Cisco’s premium customers, with an emphasis on network routing protocols.

With any network deployment, it becomes important to understand and learn how to troubleshoot the network and the technologies the network uses. Organizations strive to achieve five 9s (that is, 99.999%) availability of their network. This makes it more important that the network engineers attain the skills to troubleshoot such complex network environments. BGP has features that provide such a highly available network that some large hosting companies use only BGP. This book delivers a convenient reference for troubleshooting, deployment of best practices, and advanced protocol theory of BGP.

Joseph Pinto

SVP, Technical Services

Cisco, San Jose
Introduction

BGP is a standardized routing protocol that provides scalability, flexibility, and network stability for a variety of functions. Originally, BGP was developed to support large IP routing tables. It is the de facto protocol for routers connecting to the Internet, which provides connectivity to more than 600,000 networks and continues to grow.

Although BGP provides scalability and unique routing policy, the architecture can be intimidating or create complexity, too. Over the years, BGP has had significant increases in functionality and feature enhancements. BGP has expanded from being an Internet routing protocol to other aspects of the network, including the data center. BGP provides a scalable control plane for IPv6, MPLS VPNs (L2 and L3), Multicast, VPLS, and Ethernet VPN (EVPN).

Although most network engineers understand how to configure BGP, they lack the understanding to effectively troubleshoot BGP issues. This book is the single source for mastering techniques to troubleshoot all BGP issues for the following Cisco operating systems: Cisco IOS, IOS XR, and NX-OS. Bringing together content previously spread across multiple sources and Cisco Press titles, it covers updated various BGP design implementations found in blended service providers and enterprise environments and how to troubleshoot them.

Who Should Read This Book?

This book is for network engineers, architects, or consultants who want to learn more about BGP and learn how to troubleshoot all the various capabilities and features that it provides. Readers should have a fundamental understanding of IP routing.

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.

Part I, “BGP Fundamentals,” provides an overview of BGP fundamentals—its various attributes and features.

- Chapter 1, “BGP Fundamentals”: This chapter provides a brief overview of the BGP protocols, configuration, and some of the most commonly used features. Additional information is provided on how BGP’s behavior is different between an internal and an external BGP neighbor.

Part II, “Common BGP Troubleshooting,” provides the basic building blocks for troubleshooting BGP. These concepts are then carried over into other sections of the book.
Chapter 2, “Generic Troubleshooting Methodologies”: This chapter discusses the various basic troubleshooting methodologies and tools that are used for troubleshooting generic network problems. It also discusses how to approach a problem and how the problem can be replicated to identify the root cause.

Chapter 3, “Troubleshooting Peering Issues”: This chapter discusses the common issues seen with BGP peering. It provides detailed troubleshooting methods that can be used when investigating BGP peering issues, such as peer down and peer flapping. The chapter finally concludes by discussing dynamic BGP peering functionality.

Chapter 4, “Troubleshooting Route Advertisement and BGP Policies”: This chapter covers the BGP path selection mechanism and troubleshooting complex BGP path selection or missing route issues, which are commonly seen in BGP deployments.

Chapter 5, “Troubleshooting BGP Convergence”: This chapter examines various scenarios and conditions that could cause convergence issues. It provides a detailed explanation of how the BGP messages are formatted for the update and the complete update generation process on all the platforms.

Part III, “BGP Scalability Issues,” explains how specific problems can arise in a scaled BGP network.

Chapter 6, “Troubleshooting Platform Issues Due to BGP”: This chapter examines various platform issues that are usually seen in a production environment caused by BGP. It examines conditions such as high CPU conditions, high memory utilization, and memory leak conditions caused by BGP.

Chapter 7, “Scaling BGP”: This chapter walks you through various features in BGP that can be implemented to scale the BGP environment. It explains in detail how to scale BGP using route reflectors and other advanced features, such as BGP diverse paths.

Chapter 8, “Troubleshooting BGP Edge Architectures”: This chapter discusses BGP multihoming, which is mostly deployed in enterprise networks. It also discusses problems faced with the multihomed deployments. This chapter also explains how to achieve load balancing with BGP and how to troubleshoot any problems faced with such deployments.

Part IV, “Securing BGP,” discusses how BGP can be secured and how BGP can be used to prevent attacks in the network.

Chapter 9, “Securing BGP”: This chapter explains various features that help to secure Internet routing and thus prevent outages due to security breaches. It explains and differentiates between S-BGP and SO-BGP. The chapter then explains the SIDR solution using RPKI. Then we talk about DDoS attacks and mitigating them through RTBH and the BGP Flowspec feature.

Part V, “Multiprotocol BGP,” discusses Multiprotocol BGP and how other address families provide connectivity outside traditional IP routing.
■ **Chapter 10, “MPLS Layer 3 VPN (L3VPN)”**: This chapter discusses and explains various BGP use cases of Multi-Protocol BGP deployment in Layer 3 MPLS VPN services and how to troubleshoot them. It also describes how to scale the network in the service provider environment for L3 VPN services.

■ **Chapter 11, “BGP for MPLS L2VPN Services”**: This chapter discusses and explains various BGP use cases of Multi-Protocol BGP deployment in Layer 2 MPLS VPN services and how to troubleshoot them. It talks about features such as BGP autodiscovery for VPLS and EVPN.

■ **Chapter 12, “IPv6 BGP for Service Providers”**: This chapter covers various IPv6 services for service providers, such as 6PE, 6VPE, and methods for how to troubleshoot the problems with such deployments.

■ **Chapter 13, “VxLAN BGP EVPN”**: This chapter covers implementation of BGP in data-center deployments by providing VxLAN Overlay using BGP. The chapter also explains how the VxLAN BGP EVPN control-plane learning mechanism works and how to troubleshoot various issues faced with the VxLAN EVPN feature.

Part VI, “High Availability,” explains the techniques to increase the availability of BGP in the network.

■ **Chapter 14, “BGP High Availability”**: High availability is one of the primary concerns in almost all network deployments. This chapter discusses in detail the various high-availability features such as GR, NSR, BFD, and so on that can be implemented in BGP.

Part VII, “BGP: Looking Forward,” provides an overview of the recent enhancements to BGP and insight into future applications of BGP.

■ **Chapter 15, “Enhancements in BGP”**: This chapter discusses new enhancements in BGP, such as BGP for Link-State distribution, BGP for tunnel setup, and EVPN.

---

**Learning in a Lab Environment**

This book may contain new features and functions that do not match your current environment. As with any new technology, it is best to test in advance of actual deployment of new features.

Cisco Virtual Internet Routing Lab (VIRL) provides a scalable, extensible network design and simulation environment. Many customers use VIRL for a variety of testing before deployment of features or verification of the techniques explained in this book. VIRL includes several Cisco Network Operating System virtual machines (IOSv, IOS-XRv, CSR1000v, NX-OSv, IOSvL2, and ASAv) and has the capability to integrate with third-party vendor virtual machines or external network devices. It includes many unique capabilities, such as live visualization, that provide the capability to create protocol diagrams in real-time from your running simulation. More information about VIRL can be found at http://virl.cisco.com.
Additional Reading

The authors tried to keep the size of the book manageable while providing only necessary information for the topics involved.

Some readers may require additional reference material around the design concepts using BGP and may find the following books a great supplementary resource for the topics in this book:


The following topics are covered in this chapter:

- BGP Messages and Inter-Router Communication
- Basic BGP Configuration for IOS, IOS XR, and NX-OS
- IBGP Rules
- EBGP Rules
- BGP Route Aggregation

A router’s primary function is to move packets from one network to a different network. A router learns about unattached networks through static configuration or through dynamic routing protocols that distribute network topology information between routers. Routers try to select the best loop-free path in a network based on the destination network. Link flaps, router crashes, and other unexpected events could impact the best path, so the routers must exchange information with each other so that the network topology updates during these types of events.

Routing protocols are classified as either an Interior Gateway Protocol (IGP) or an Exterior Gateway Protocol (EGP), which indicates whether the protocol is designed for exchanging routes within an organization or between organizations. In IGP protocols, all routers use a common logic within the routing domain to find the shortest path to reach a destination. EGP protocols may require a unique routing policy for every external organization that it exchanges routes.

**Border Gateway Protocol**

RFC 1654 defines Border Gateway Protocol (BGP) as an EGP standardized path-vector routing protocol that provides scalability, flexibility, and network stability. When BGP was created, the primary design consideration was for IPv4 inter-organization
connectivity on public networks, such as the Internet, or private dedicated networks. BGP is the only protocol used to exchange networks on the Internet, which has more than 600,000 IPv4 routes and continues to grow. BGP does not advertise incremental updates or refresh network advertisements like OSPF or ISIS. BGP prefers stability within the network, because a link flap could result in route computation for thousands of routes.

From the perspective of BGP, an autonomous system (AS) is a collection of routers under a single organization's control, using one or more IGPs, and common metrics to route packets within the AS. If multiple IGPs or metrics are used within an AS, the AS must appear consistent to external ASs in routing policy. An IGP is not required within an AS, and could use BGP as the only routing protocol in it, too.

**Autonomous System Numbers**

Organizations requiring connectivity to the Internet must obtain an Autonomous System Number (ASN). ASNs were originally 2 bytes (16 bit) providing 65,535 ASNs. Due to exhaustion, RFC 4893 expands the ASN field to accommodate 4 bytes (32 bit). This allows for 4,294,967,295 unique ASNs, providing quite a leap from the original 65,535 ASNs.

Two blocks of private ASNs are available for any organization to use as long as they are never exchanged publicly on the Internet. ASNs 64,512–65,535 are private ASNs within the 16-bit ASN range, and 4,200,000,000–4,294,967,294 are private ASNs within the extended 32-bit range.

The Internet Assigned Numbers Authority (IANA) is responsible for assigning all public ASNs to ensure that they are globally unique. IANA requires the following items when requesting a public ASN:

- Proof of a publicly allocated network range
- Proof that Internet connectivity is provided through multiple connections
- Need for a unique route policy from your providers

In the event that an organization does not meet those guidelines, it should use the ASN provided by its service provider.

**Note**  It is imperative that you use only the ASN assigned by IANA, the ASN assigned by your service provider, or private ASNs. Using another organization's ASN without permission could result in traffic loss and cause havoc on the Internet.
Path Attributes

BGP attaches path attributes (PA) associated with each network path. The PAs provide BGP with granularity and control of routing policies within BGP. The BGP prefix PAs are classified as follows:

- Well-known mandatory
- Well-known discretionary
- Optional transitive
- Optional nontransitive

Per RFC 4271, well-known attributes must be recognized by all BGP implementations. Well-known mandatory attributes must be included with every prefix advertisement, whereas well-known discretionary attributes may or may not be included with the prefix advertisement.

Optional attributes do not have to be recognized by all BGP implementations. Optional attributes can be set so that they are transitive and stay with the route advertisement from AS to AS. Other PAs are nontransitive and cannot be shared from AS to AS. In BGP, the Network Layer Reachability Information (NLRI) is the routing update that consists of the network prefix, prefix length, and any BGP PAs for that specific route.

Loop Prevention

BGP is a path vector routing protocol and does not contain a complete topology of the network-like link state routing protocols. BGP behaves similar to distance vector protocols to ensure a path is loop free.

The BGP attribute AS_PATH is a well-known mandatory attribute and includes a complete listing of all the ASNs that the prefix advertisement has traversed from its source AS. The AS_PATH is used as a loop prevention mechanism in the BGP protocol. If a BGP router receives a prefix advertisement with its AS listed in the AS_PATH, it discards the prefix because the router thinks the advertisement forms a loop.

Address Families

Originally, BGP was intended for routing of IPv4 prefixes between organizations, but RFC 2858 added Multi-Protocol BGP (MP-BGP) capability by adding extensions called address-family identifier (AFI). An address-family correlates to a specific network protocol, such as IPv4, IPv6, and the like, and additional granularity through a subsequent address-family identifier (SAFI), such as unicast and multicast. MBGP achieves this separation by using the BGP path attributes (PAs) MP_REACH_NLRI and MP_UNREACH_NLRI. These attributes are carried inside BGP update messages and are used to carry network reachability information for different address families.
Network engineers and vendors continue to add functionality and feature enhancements to BGP. BGP now provides a scalable control plane for signaling for overlay technologies like MPLS VPNs, IPsec Security Associations, and Virtual Extensible LAN (VXLAN). These overlays can provide Layer 3 connectivity via MPLS L3VPNs, or Layer 2 connectivity via MPLS L2VPNs (L2VPN), such as Virtual Private LAN Service (VPLS) or Ethernet VPNs (EVPNs).

Every address-family maintains a separate database and configuration for each protocol (address-family + subaddress family) in BGP. This allows for a routing policy in one address-family to be different from a routing policy in a different address family even though the router uses the same BGP session to the other router. BGP includes an AFI and a SAFI with every route advertisement to differentiate between the AFI and SAFI databases. Table 1-1 provides a small list of common AFI and SAFIs.

<table>
<thead>
<tr>
<th>AFI</th>
<th>SAFI</th>
<th>Network Layer Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>IPv4 Unicast</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>IPv4 Multicast</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>IPv4 Unicast with MPLS Label</td>
</tr>
<tr>
<td>1</td>
<td>128</td>
<td>MPLS L3VPN IPv4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>IPv6 Unicast</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>IPv6 Unicast with MPLS Label</td>
</tr>
<tr>
<td>2</td>
<td>128</td>
<td>MPLS L3VPN IPv6</td>
</tr>
<tr>
<td>25</td>
<td>65</td>
<td>Virtual Private LAN Service (VPLS)</td>
</tr>
<tr>
<td>25</td>
<td>70</td>
<td>Virtual Private Wire Service (VPWS)</td>
</tr>
<tr>
<td>25</td>
<td>70</td>
<td>Ethernet VPN (EVPN)</td>
</tr>
</tbody>
</table>

**BGP Sessions**

A BGP session refers to the established adjacency between two BGP routers. BGP sessions are always point-to-point and are categorized into two types:

- **Internal BGP (IBGP)**: Sessions established with an IBGP router that are in the same AS or participate in the same BGP confederation. IBGP sessions are considered more secure, and some of BGP’s security measures are lowered in comparison to EBGP.
Inter-Router Communication

sessions. IBGP prefixes are assigned an administrative distance (AD) of 200 upon installing into the router’s routing information base (RIB).

- **External BGP (EBGP):** Sessions established with a BGP router that are in a different AS. EBGP prefixes are assigned an AD of 20 upon installing into the router’s RIB.

**Note**  
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.

**Inter-Router Communication**

BGP does not use hello packets to discover neighbors like IGP protocols and cannot discover neighbors dynamically. BGP was designed as an interautonomous routing protocol, implying that neighbor adjacencies should not change frequently and are coordinated. BGP neighbors are defined by an IP address.

BGP uses TCP port 179 to communicate with other routers. TCP allows for handling of fragmentation, sequencing, and reliability (acknowledgement and retransmission) of communication packets.

IGP protocols follow the physical topology because the sessions are formed with hellos that cannot cross network boundaries (that is, single hop only). BGP uses TCP, which is capable of crossing network boundaries (that is, multihop capable). While BGP can form neighbor adjacencies that are directly connected, it can also form adjacencies that are multiple hops away. Multihop sessions require that the router use an underlying route installed in the RIB (static or from any routing protocol) to establish the TCP session with the remote endpoint.

In Figure 1-1, R1 is able to establish a direct BGP session with R2. In addition, R2 is able to form a BGP session with R4, even though it passes through R3. R1 and R2 use a directly connected route to locate each other. R2 uses a static route to reach the 10.1.34.0/24 network, and R4 has a static route to reach the 10.1.23.0/24 network. R3 is unaware that R2 and R4 have established a BGP session, even though the packets flow through R3.

![Figure 1-1  BGP Direct and Multihop Sessions](image-url)
Note BGP neighbors connected via the same network use the ARP table to locate the Layer 2 address of the peer. Multihop BGP sessions require route table information for finding the IP address of the peer. It is common to have a static route or IGP running between IBGP neighbors for providing the topology path information for establishing the BGP TCP session. A default route is not sufficient to form a multihop BGP session.

BGP can be thought of as a control plane routing protocol or as an application, because it allows for the exchanging of routes with peers multiple hops away. BGP routers do not have to be in the data plane (path) to exchange prefixes, but all routers in the data path need to know all the routes that will be forwarded through them.

BGP Messages

BGP communication uses four message types, as shown in Table 1-2.

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Functional Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OPEN</td>
<td>Sets up and establishes BGP adjacency</td>
</tr>
<tr>
<td>2</td>
<td>UPDATE</td>
<td>Advertises, updates, or withdraws routes</td>
</tr>
<tr>
<td>3</td>
<td>NOTIFICATION</td>
<td>Indicates an error condition to a BGP neighbor</td>
</tr>
<tr>
<td>4</td>
<td>KEEPALIVE</td>
<td>Ensures that BGP neighbors are still alive</td>
</tr>
</tbody>
</table>

OPEN

The OPEN message is used to establish a BGP adjacency. Both sides negotiate session capabilities before a BGP peering establishes. The OPEN message contains the BGP version number, ASN of the originating router, Hold Time, BGP Identifier, and other optional parameters that establish the session capabilities.

Hold Time

The Hold Time attribute sets the Hold Timer in seconds for each BGP neighbor. Upon receipt of an UPDATE or KEEPALIVE, the Hold Timer resets to the initial value. If the Hold Timer reaches zero, the BGP session is torn down, routes from that neighbor are removed, and an appropriate update route withdraw message is sent to other BGP neighbors for the impacted prefixes. The Hold Time is a heartbeat mechanism for BGP neighbors to ensure that the neighbor is healthy and alive.

When establishing a BGP session, the routers use the smaller Hold Time value contained in the two router’s OPEN messages. The Hold Time value must be at least three seconds, or zero. For Cisco routers the default hold timer is 180 seconds.
BGP Identifier

The BGP Router-ID (RID) is a 32-bit unique number that identifies the BGP router in the advertised prefixes as the BGP Identifier. The RID can be used as a loop prevention mechanism for routers advertised within an autonomous system. The RID can be set manually or dynamically for BGP. A nonzero value must be set for routers to become neighbors. The dynamic RID allocation logic varies between the following operating systems.

- **IOS**: IOS nodes use the highest IP address of the any *up* loopback interfaces. If there is not an *up* loopback interface, then the highest IP address of any active *up* interfaces becomes the RID when the BGP process initializes.
- **IOS XR**: IOS XR nodes use the IP address of the lowest *up* loopback interface. If there is not any *up* loopback interfaces, then a value of zero (0.0.0.0) is used and prevents any BGP adjacencies from forming.
- **NX-OS**: NX-OS nodes use the IP address of the lowest *up* loopback interface. If there is not any *up* loopback interfaces, then the IP address of the lowest active *up* interface becomes the RID when the BGP process initializes.

Router-IDs typically represent an IPv4 address that resides on the router, such as a loopback address. Any IPv4 address can be used, including IP addresses not configured on the router. For IOS and IOS XR, the command `bgp router-id router-id` is used, and NX-OS uses the command `router-id router-id` under the BGP router configuration to statically assign the BGP RID. Upon changing the router-id, all BGP sessions reset and need to be reestablished.

**Note**  Setting a static BGP RID is a best practice.

**KEEPALIVE**

BGP does not rely on the TCP connection state to ensure that the neighbors are still alive. Keepalive messages are exchanged every one-third of the Hold Timer agreed upon between the two BGP routers. Cisco devices have a default Hold Time of 180 seconds, so the default Keepalive interval is 60 seconds. If the Hold Time is set for zero, no Keepalive messages are sent between the BGP neighbors.

**UPDATE**

The Update message advertises any feasible routes, withdraws previously advertised routes, or can do both. The Update message includes the Network Layer Reachability Information (NLRI) that includes the prefix and associated BGP PAs when advertising prefixes. Withdrawn NLRLs include only the prefix. An UPDATE message can act as a Keepalive to reduce unnecessary traffic.
NOTIFICATION Message

A Notification message is sent when an error is detected with the BGP session, such as a hold timer expiring, neighbor capabilities change, or a BGP session reset is requested. This causes the BGP connection to close.

BGP Neighbor States

BGP forms a TCP session with neighbor routers called peers. BGP uses the Finite State Machine (FSM) to maintain a table of all BGP peers and their operational status. The BGP session may report in the following states:

- Idle
- Connect
- Active
- OpenSent
- OpenConfirm
- Established

Figure 1-2 displays the BGP FSM and the states in order of establishing a BGP session.

Figure 1-2  BGP Finite State Machine
Idle

This is the first stage of the BGP FSM. BGP detects a start event, tries to initiate a TCP connection to the BGP peer, and also listens for a new connect from a peer router.

If an error causes BGP to go back to the Idle state for a second time, the ConnectRetryTimer is set to 60 seconds and must decrement to zero before the connection is initiated again. Further failures to leave the Idle state result in the ConnectRetryTimer doubling in length from the previous time.

Connect

In this state, BGP initiates the TCP connection. If the 3-way TCP handshake completes, the established BGP Session BGP process resets the ConnectRetryTimer and sends the Open message to the neighbor, and then changes to the OpenSent State.

If the ConnectRetry timer depletes before this stage is complete, a new TCP connection is attempted, the ConnectRetry timer is reset, and the state is moved to Active. If any other input is received, the state is changed to Idle.

During this stage, the neighbor with the higher IP address manages the connection. The router initiating the request uses a dynamic source port, but the destination port is always 179.

Example 1-1 shows an established BGP session using the command `show tcp brief` to display the active TCP sessions between routers. Notice that the TCP source port is 179 and the destination port is 59884 on R1, and the ports are opposite on R2.

**Example 1-1  Established BGP Session**

```
RP/0/0/CPU0:R1# show tcp brief | exc "LISTEN|CLOSED"
   PCB  VRF-ID  Recv-Q  Send-Q  Local Address    Foreign Address     State
0x088bcbb8 0x60000000      0      0  10.1.12.1:179    10.1.12.2:59884    ESTAB

R2# show tcp brief
   TCB       Local Address               Foreign Address             (state)
EF153B88  10.1.12.2.59884             10.1.12.1.179                ESTAB
```

**Note**  Service providers consistently assign their customers the higher or lower IP address for their networks. This helps the service provider create proper instructions for access control lists (ACL) or firewall rules, or for troubleshooting them.
**Active**

In this state, BGP starts a new 3-way TCP handshake. If a connection is established, an Open message is sent, the Hold Timer is set to 4 minutes, and the state moves to OpenSent. If this attempt for TCP connection fails, the state moves back to the Connect state and resets the ConnectRetryTimer.

**OpenSent**

In this state, an Open message has been sent from the originating router and is awaiting an Open message from the other router. After the originating router receives the OPEN message from the other router, both OPEN messages are checked for errors. The following items are being compared:

- BGP Versions must match.
- The source IP address of the OPEN message must match the IP address that is configured for the neighbor.
- The AS number in the OPEN message must match what is configured for the neighbor.
- BGP Identifiers (RID) must be unique. If a RID does not exist, this condition is not met.
- Security Parameters (Password, TTL, and the like).

If the Open messages do not have any errors, the Hold Time is negotiated (using the lower value), and a KEEPALIVE message is sent (assuming the value is not set to zero). The connection state is then moved to OpenConfirm. If an error is found in the OPEN message, a Notification message is sent, and the state is moved back to Idle.

If TCP receives a disconnect message, BGP closes the connection, resets the ConnectRetryTimer, and sets the state to Active. Any other input in this process results in the state moving to Idle.

**OpenConfirm**

In this state, BGP waits for a Keepalive or Notification message. Upon receipt of a neighbor’s Keepalive, the state is moved to Established. If the hold timer expires, a stop event occurs, or a Notification message is received, and the state is moved to Idle.

**Established**

In this state, the BGP session is established. BGP neighbors exchange routes via Update messages. As Update and Keepalive messages are received, the Hold Timer is reset. If the Hold Timer expires, an error is detected and BGP moves the neighbor back to the Idle state.
Basic BGP Configuration

When configuring BGP, it is best to think of the configuration from a modular perspective. BGP router configuration requires the following components:

- **BGP Session Parameters**: BGP session parameters provide settings that involve establishing communication to the remote BGP neighbor. Session settings include the ASN of the BGP peer, authentication, and keepalive timers.

- **Address-Family Initialization**: The address-family is initialized under the BGP router configuration mode. Networks advertisement and summarization occur within the address-family.

- **Activate the Address-Family on the BGP Peer**: Activate the address-family on the BGP peer. For a session to initiate, one address-family for that neighbor must be activated. The router's IP address is added to the neighbor table, and BGP attempts to establish a BGP session or accepts a BGP session initiated from the peer router.

For the remainder of this chapter, the BGP context is directed toward IPv4 routing. Other address families are throughout the book.

**IOS**

The steps for configuring BGP on an IOS router are as follows:

**Step 1.** Create the BGP Routing Process. Initialize the BGP process with the global command `router bgp as-number`.

**Step 2.** Identify the BGP Neighbor’s IP address and Autonomous System Number. Identify the BGP neighbor’s IP address and autonomous system number with the BGP router configuration command `neighbor ip-address remote-as as-number`.

**Note**  
IOS activates the IPv4 address-family by default. This can simplify the configuration in an IPv4 environment because Steps 3 and 4 are optional, but may cause confusion when working with other address families. The BGP router configuration command `no bgp default ip4-unicast` disables the automatic activation of the IPv4 AFI so that Steps 3 and 4 are required.

**Step 3.** Initialize the address-family with the BGP router configuration command `address-family afi safi`.

**Step 4.** Activate the address-family for the BGP neighbor with the BGP address-family configuration command `neighbor ip-address activate`. 
Note  On IOS routers, the default address-family modifier for the IPv4 and IPv6 address families is unicast and is optional. The address-family modifier is required on IOS XR nodes.

Example 1-2 demonstrates how to configure R1 and R2 using the IOS default and optional IPv4 AFI modifier CLI syntax. R1 is configured using the default IPv4 address-family enabled, and R2 disables IOS’s default IPv4 address-family and manually activates it for the specific neighbor 10.1.12.1.

Example 1-2  IOS Basic BGP Configuration

<table>
<thead>
<tr>
<th>R1 (Default IPv4 Address-Family Enabled)</th>
<th>R2 (Default IPv4 Address-Family Disabled)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>router bgp 65100</code></td>
<td><code>router bgp 65100</code></td>
</tr>
<tr>
<td><code>neighbor 10.1.12.2 remote-as 65100</code></td>
<td><code>no bgp default ipv4-unicast</code></td>
</tr>
<tr>
<td></td>
<td><code>neighbor 10.1.12.1 remote-as 65100</code></td>
</tr>
<tr>
<td></td>
<td><code>!</code></td>
</tr>
<tr>
<td></td>
<td><code>address-family ipv4</code></td>
</tr>
<tr>
<td></td>
<td><code>neighbor 10.1.12.1 activate</code></td>
</tr>
<tr>
<td></td>
<td><code>exit-address-family</code></td>
</tr>
</tbody>
</table>

IOS XR

The steps for configuring BGP on an IOS XR router are as follows:

Step 1. Create the BGP routing process. Initialize the BGP process with the global configuration command `router bgp as-number`.

Step 2. Initialize the address-family with the BGP router configuration command `address-family afi safi` so it can be associated to a BGP neighbor.

Step 3. Identify the BGP neighbor’s IP address with the BGP router configuration command `neighbor ip-address`.

Step 4. Identify the BGP neighbor’s autonomous system number with the BGP neighbor configuration command `remote-as as-number`.

Step 5. Activate the address-family for the BGP neighbor with the BGP neighbor configuration command `address-family afi safi`.

Step 6. Associate a route policy for EBGP Peers. IOS XR requires a routing policy to be associated to an EBGP peer as a security measure to ensure that routes are not accidentally accepted or advertised. If a route policy is not configured in
the appropriate address-family, then NLRIs are discarded upon receipt and no NLRIs are advertised to EBGP peers.

An inbound and outbound route policy is configured with the command `route-policy policy-name (in | out)` under the BGP neighbor address-family configuration.

**Note** IOS XR nodes do not establish a BGP session if the RID is set to zero, because the dynamic RID allocation did not find any up loopback interfaces. The RID needs to be set manually with the BGP router configuration command `bgp router-id`.

Example 1-3 displays the BGP configuration for R1 if it was running IOS XR. The RID is set on R1 because that router does not have any loopback interfaces.

**Example 1-3  IOS XR BGP Configuration**

```
IOS XR
router bgp 65100
  bgp router-id 192.168.1.1
  address-family ipv4 unicast

neighbor 10.1.12.2
  remote-as 65100
  address-family ipv4 unicast
```

**NX-OS**

The steps for configuring BGP on an NX-OS device are as follows:

**Step 1.** Create the BGP routing process. Initialize the BGP process with the global configuration command `router bgp as-number`.

**Step 2.** Initialize the address-family with the BGP router configuration command `address-family afi safi` so it can be associated to a BGP neighbor.

**Step 3.** Identify the BGP neighbor’s IP address and autonomous system number with the BGP router configuration command `neighbor ip-address remote-as as-number`.

**Step 4.** Activate the address-family for the BGP neighbor with the BGP neighbor configuration command `address-family afi safi`.
Example 1-4 displays the BGP configuration for R1 if it was running NX-OS.

**Example 1-4  NX-OS BGP Configuration**

```plaintext
NX-OS
router bgp 65100
  address-family ipv4 unicast
  neighbor 10.1.12.2 remote-as 65100
  address-family ipv4 unicast
```

**Verification of BGP Sessions**

The BGP session is verified with the command `show bgp afi safi summary` on IOS, IOS XR, and NX-OS devices. Example 1-5 displays the IPv4 BGP unicast summary. Notice that the BGP RID and table versions are the first components shown. The Up/Down column reflects that the BGP session is up for over 5 minutes.

**Example 1-5  BGP IPv4 Session Summary Verification**

```plaintext
R1-IOS# show bgp ipv4 unicast summary
BGP router identifier 192.168.2.2, local AS number 65100
BGP table version is 1, main routing table version 1

Neighbor  V  AS MsgRcvd MsgSent  TblVer  InQ OutQ  Up/Down  State/PfxRcd
10.1.12.2 4  65100       8       9        1    0    0 00:05:23        0

RP/0/CPU0:R1-XR# show bgp ipv4 unicast summary
! Output omitted for brevity
BGP router identifier 192.168.1.1, local AS number 65100
BGP main routing table version 4

Process   RcvTblVer  bRIB/RIB  LabelVer  ImportVer  SendTblVer  StandByVer
Speaker             4          4          4          4           4           4

Neighbor  Spk  AS MsgRcvd MsgSent  TblVer  InQ OutQ  Up/Down  St/PfxRcd
10.1.12.2 0 65100      32      37        5    0    0 00:05:24          0

R1-NXOS# show bgp ipv4 unicast summary
! Output omitted for brevity
BGP router identifier 192.168.1.1, local AS number 65100
BGP table version is 5, IPv4 Unicast config peers 2, capable peers 1

Neighbor  V  AS MsgRcvd MsgSent  TblVer  InQ OutQ  Up/Down  State/PfxRcd
10.1.12.2 4  65100      32      37        5    0    0 00:05:24          0
```
Table 1-3 explains the fields of output when displaying the BGP Table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighbor</td>
<td>IP address of the BGP peer</td>
</tr>
<tr>
<td>V</td>
<td>BGP Version spoken by BGP peer (IOS and NX-OS only)</td>
</tr>
<tr>
<td>AS</td>
<td>Autonomous system number of BGP peer</td>
</tr>
<tr>
<td>MsgRcvd</td>
<td>Count of messages received from the BGP peer</td>
</tr>
<tr>
<td>MsgSent</td>
<td>Count of messages sent to the BGP peer</td>
</tr>
<tr>
<td>TblVer</td>
<td>Last version of the BGP database sent to the peer</td>
</tr>
<tr>
<td>InQ</td>
<td>Number of messages queued to be processed from the peer</td>
</tr>
<tr>
<td>OutQ</td>
<td>Number of messages queued to be sent to the peer</td>
</tr>
<tr>
<td>Up/Down</td>
<td>Length of time the BGP session is established, or the current status if the</td>
</tr>
<tr>
<td></td>
<td>session is not in established state</td>
</tr>
<tr>
<td>State/PfxRcd</td>
<td>Current state of BGP peer or the number of prefixes received from the peer</td>
</tr>
</tbody>
</table>

**Note** Earlier commands like `show ip bgp summary` came out before MBGP and do not provide a structure for the current multiprotocol capabilities within BGP. Using the AFI and SAFI syntax ensures consistency for the commands regardless of information exchanged by BGP.

BGP neighbor session state, timers, and other essential peering information is shown with the command `show bgp afi safi neighbors ip-address`, as shown in Example 1-6.

**Example 1-6** BGP IPv4 Neighbor Output

```
R2# show bgp ipv4 unicast neighbors 10.1.12.1
! Output omitted for brevity

! The first section provides the neighbor's IP address, remote-as, indicates if
! the neighbor is 'internal' or 'external', the neighbor's BGP version, RID,
! session state, and timers.
BGP neighbor is 10.1.12.1, remote AS100, internal link
BGP version 4, remote router ID 192.168.1.1
BGP state = Established, up for 00:01:04
Last read 00:00:10, last write 00:00:09, hold is 180, keepalive is 60 seconds
Neighbor sessions:
  1 active, is not multisession capable (disabled)
```
Chapter 1: BGP Fundamentals

This second section indicates the capabilities of the BGP neighbor and address-families configured on the neighbor.

Neighbor capabilities:
- Route refresh: advertised and received (new)
- Four-octets ASN Capability: advertised and received
- Address family IPv4 Unicast: advertised and received
- Enhanced Refresh Capability: advertised
- Multisession Capability:
  - Stateful switchover support enabled: NO for session 1

Message statistics:
- InQ depth is 0
- OutQ depth is 0

This section provides a list of the BGP packet types that have been received or sent to the neighbor router.

<table>
<thead>
<tr>
<th></th>
<th>Sent</th>
<th>Rcvd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opens:</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Notifications:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Updates:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Keepalives:</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Route Refresh:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total:</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Default minimum time between advertisement runs is 0 seconds

This section provides the BGP table version of the IPv4 Unicast address-family. The table version is not a 1-to-1 correlation with routes as multiple route change can occur during a revision change. Notice the Prefix Activity columns in this section.

For address family: IPv4 Unicast
- Session: 10.1.12.1
- BGP table version 1, neighbor version 1/0
- Output queue size : 0
- Index 1, Advertise bit 0

<table>
<thead>
<tr>
<th></th>
<th>Sent</th>
<th>Rcvd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix activity:</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>Prefixes Current:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Prefixes Total:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Implicit Withdraw:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Explicit Withdraw:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Used as bestpath:</td>
<td>n/a</td>
<td>0</td>
</tr>
<tr>
<td>Used as multipath:</td>
<td>n/a</td>
<td>0</td>
</tr>
</tbody>
</table>

Outbound Inbound
- Local Policy Denied Prefixes: ------- -------
- Total: 0    0

Number of NLRIs in the update sent: max 0, min 0
This section indicates that a valid route exists in the RIB to the BGP peer IP address, provides the number of times that the connection has established and time dropped, since the last reset, the reason for the reset, if path-mtu-discovery is enabled, and ports used for the BGP session.

Address tracking is enabled, the RIB does have a route to 10.1.12.1
Connections established 2, dropped 1
Last reset 00:01:40, due to Peer closed the session
Transport(tcp) path-mtu-discovery is enabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Minimum incoming TTL 0, Outgoing TTL 255
Local host: 10.1.12.2, Local port: 179
Foreign host: 10.1.12.1, Foreign port: 56824

Prefix Advertisement

BGP uses three tables for maintaining the network prefix and path attributes (PA) for a route. The BGP tables are as follows:

- **Adj-RIB-in**: Contains the NLRIs in original form before inbound route policies are processed. The table is purged after all route policies are processed to save memory.

- **Loc-RIB**: Contains all the NLRIs that originated locally or were received from other BGP peers. After NLRIs pass the validity and next-hop reachability check, the BGP best path algorithm selects the best NLRI for a specific prefix. The Loc-RIB table is the table used for presenting routes to the ip routing table.

- **Adj-RIB-out**: Contains the NLRIs after outbound route policies have processed.

BGP `network` statements do not enable BGP for a specific interface. Instead they identify a specific network prefix to be installed into the BGP table, known as the **Loc-RIB table**.

After configuring a BGP `network` statement, the BGP process searches the global RIB for an exact network prefix match. The network prefix can be a connected network, secondary connected network, or any route from a routing protocol. After verifying that the network statement matches a prefix in the global RIB, the prefix installs into the BGP Loc-RIB table. As the BGP prefix installs into the Loc-RIB, the following BGP PA are set depending on the RIB prefix type:

- **Connected Network**: The next-hop BGP attribute is set to 0.0.0.0, the origin attribute is set to `i (IGP)`, and the BGP weight is set to 32,768.

- **Static Route or Routing Protocol**: The next-hop BGP attribute is set to the next-hop IP address in the RIB, the origin attribute is set to `i (IGP)`, the BGP weight is set to 32,768; and the MED is set to the IGP metric.

The network statement resides under the appropriate address-family within the BGP router configuration. The command `network network mask subnet-mask [route-map route-map-name]` is used for advertising IPv4 networks on IOS and NX-OS devices.
NX-OS devices also support prefix-length notation with the command `network network/prefix-length [route-map route-map-name]`. IOS XR routers use the command `network network/prefix-length [route-policy route-policy-name]` for installing routes into the BGP table. The optional `route-map` or `route-policy` parameter provides a method to set specific BGP PAs when the prefix installs into the Loc-RIB.

The command `show bgp afi safi` displays the contents of the BGP database (Loc-RIB) on IOS, IOS XR, and NX-OS devices. Every entry in the BGP Loc-RIB table contains at least one route, but could contain multiple routes for the same network prefix.

**Note**  By default, BGP advertises only the best path to other BGP peers regardless of the number of routes (NLRIs) in the BGP Loc-RIB. The BGP best path executes individually per address-family. The best path selection of one address-family cannot impact the best path calculation on a different address-family.

Example 1-7 displays the BGP table for IOS, IOS XR, and NX-OS. The BGP table contains received routes and locally generated routes.

**Example 1-7  Display of BGP Table**

```
R1-IOS# show bgp ipv4 unicast
BGP table version is 5, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter, x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

Network          Next Hop            Metric LocPrf Weight Path
*>  192.168.1.1/32   0.0.0.0                  0         32768 i
*   192.168.2.2/32   10.1.13.3                              0 65300 65200 i
*>                  10.1.12.2                0             0 65200 i
*>  192.168.3.3/32   10.1.13.3                              0 65300 i
*                    10.1.12.2                              0 65200 65300 i

RP/0/0/CPU0:R2-XR#
show bgp ipv4 unicast
! Output omitted for brevity
BGP router identifier 192.168.2.2, local AS number 65200
Status codes: s suppressed, d damped, h history, * valid, > best i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.1.1/32   10.1.12.1                0             0 65100 i
*                    10.1.23.3                              0 65300 65100 i
```
Basic BGP Configuration

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.2.2/32</td>
<td>0.0.0.0</td>
<td>0</td>
<td></td>
<td>32768</td>
<td>i</td>
</tr>
<tr>
<td>192.168.3.3/32</td>
<td>10.1.12.1</td>
<td>0</td>
<td>0</td>
<td>65100</td>
<td>65300 i</td>
</tr>
<tr>
<td>10.1.23.3</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>65300 i</td>
</tr>
</tbody>
</table>

Processed 3 prefixes, 5 paths

R3-NXOS# show bgp ipv4 unicast

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.1/32</td>
<td>10.1.13.1</td>
<td>0</td>
<td></td>
<td>0</td>
<td>65100 i</td>
</tr>
<tr>
<td>192.168.2.2/32</td>
<td>10.1.23.2</td>
<td>0</td>
<td></td>
<td>0</td>
<td>65200 65100 i</td>
</tr>
<tr>
<td>192.168.3.3/32</td>
<td>0.0.0.0</td>
<td>100</td>
<td></td>
<td>32768</td>
<td>i</td>
</tr>
</tbody>
</table>

Note: NX-OS devices place e beside external learned BGP routes and l beside locally advertised BGP routes. IOS and IOS XR devices do not have this behavior.

Table 1-4 explains the fields of output when displaying the BGP table.

**Table 1-4 BGP Table Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>List of the network prefixes installed in BGP. If multiple NLRIs exist for the same prefix, only the first prefix is identified, and others leave a blank space. Valid NLRIs are indicated by the *. The NLRI selected as the best path is indicated by an angle bracket (&gt;).</td>
</tr>
<tr>
<td>Next Hop</td>
<td><em>Next Hop:</em> A well-known mandatory BGP path attribute that defines the IP address for the next-hop for that specific NLRI.</td>
</tr>
<tr>
<td>Metric</td>
<td><em>Multiple-Exit Discriminator (MED):</em> An optional nontransitive BGP path attribute used in BGP algorithm for that specific NLRI.</td>
</tr>
<tr>
<td>LocPrf</td>
<td><em>Local Preference:</em> A well-known discretionary BGP path attribute used in the BGP best path algorithm for that specific NLRI.</td>
</tr>
<tr>
<td>Weight</td>
<td>Locally significant Cisco defined attribute used in the BGP best path algorithm for that specific NLRI.</td>
</tr>
</tbody>
</table>
### BGP Best-Path Calculation

In BGP, route advertisements consist of the Network Layer Reachability Information (NLRI) and the path attributes (PAs). The NLRI composes the network prefix and prefix-length, and the BGP attributes such as AS-Path, Origin, and the like are stored in the path attributes. A BGP route may contain multiple paths to the same destination network. Every path's attributes impact the desirability of the route when a router selects the best path. A BGP router advertises only the best path to the neighboring routers.

Inside the BGP Loc-RIB table, all the routes and their path attributes are maintained with the best path calculated. The best path is then installed in the RIB of the router. In the event the best path is no longer available, the router can use the existing paths to quickly identify a new best path. BGP recalculates the best path for a prefix upon four possible events:

- BGP next-hop reachability change
- Failure of an interface connected to an EBGP peer
- Redistribution change
- Reception of new paths for a route

The BGP best path selection algorithm influences how traffic enters or leaves an autonomous system (AS). BGP does not use metrics to identify the best path in a network. BGP uses path attributes to identify its best path.

Some router configurations modify the BGP attributes to influence inbound traffic, outbound traffic, or inbound and outbound traffic depending on the network design requirements. BGP path attributes can be modified upon receipt or advertisement to influence routing in the local AS or neighboring AS. A basic rule for traffic engineering with BGP is that modifications in outbound routing policies influence inbound traffic, and modifications to inbound routing policies influence outbound traffic.

BGP installs the first received path as the best path automatically. When additional paths are received, the newer paths are compared against the current best path. If there is a tie, then processing continues onto the next step, until a best path winner is identified.
The following list provides the attributes that the BGP best path algorithm uses for the best route selection process. These attributes are processed in the order listed:

1. Weight
2. Local Preference
3. Local originated (network statement, redistribution, aggregation)
4. AIGP
5. Shortest-AS Path
6. Origin Type
7. Lowest MED
8. EBGP over IBGP
9. Lowest IGP Next-Hop
10. If both paths are external (EBGP), prefer the first (oldest)
11. Prefer the route that comes from the BGP peer with the lower RID
12. Prefer the route with the minimum cluster list length
13. Prefer the path that comes from the lowest neighbor address

The best path algorithm can be used to manipulate network traffic patterns for a specific route by modifying various path attributes on BGP routers. Changing of BGP PA can influence traffic flow into, out of, and around an AS.

BGP supports three types of equal cost multipath (ECMP): EBGP multipath, IBGP multipath, or eIBGP multipath. EBGP multipath requires that the weight, local preference, AS-Path length, AS-Path content, Origin, and MED match for a second route to install into the RIB. Chapter 8, “Troubleshooting BGP Edge Architectures,” explains BGP ECMP in more detail.

**Route Filtering and Manipulation**

Route filtering is a method for selectively identifying routes that are advertised or received from neighbor routers. Route filtering may be used to manipulate traffic flows, reduce memory utilization, or to improve security. For example, it is common for ISPs to deploy route filters on BGP peerings to customers. Ensuring that only the customer routes are allowed over the peering link prevents the customer from accidentally becoming a transit AS on the Internet.

Filtering of routes within BGP is accomplished with filter-lists, prefix-lists, or route-maps on IOS and NX-OS devices. IOS XR uses route policies for filtering of routes. Route-filtering is explained in more detail in Chapter 4, “Troubleshooting Route Advertisement and BGP Policies.”
Depending on the change to the BGP route manipulation technique, the BGP session may need to be refreshed to take effect. BGP supports two methods of clearing a BGP session: The first method is a hard reset, which tears down the BGP session, removes BGP routes from the peer, and is the most disruptive. The second method is a soft reset, which invalidates the BGP cache and requests a full advertisement from its BGP peer.

IOS and NX-OS devices initiate a hard reset with the command `clear ip bgp ip-address [soft]`, and the command `clear bgp ip-address [graceful]` is used on IOS XR nodes. Soft reset on IOS and NX-OS devices use the optional `soft` keyword, whereas IOS XR nodes use the optional `graceful` keyword. Sessions can be cleared with all BGP neighbors by using an asterisk `*` in lieu of the peer’s IP address.

When a BGP policy changes, the BGP table must be processed again so that the neighbors can be notified accordingly. Routes received by a BGP peer must be processed again. If the BGP session supports route refresh capability, then the peer readvertises (refreshes) the prefixes to the requesting router, allowing for the inbound policy to process using the new policy changes. The route refresh capability is negotiated for each address-family when the session is established.

Performing a soft reset on sessions that support route refresh capability actually initiates a route refresh. Soft resets can be performed for a specific address-family with the command `clear bgp address-family address-family modifier ip-address soft [in | out]`. Soft resets reduce the amount of routes that must be exchanged if multiple address families are configured with a single BGP peer. Changes to the outbound routing policies use the optional `out` keyword, and changes to inbound routing policies use the optional `in` keyword.

Older IOS versions that do not support route refresh capability require the usage of inbound soft reconfiguration so that updates to inbound route policies can be applied without performing a hard reset. Inbound soft reconfiguration does not purge the Adj-RIB-In table after routes process into the Loc-RIB table. The Adj-RIB-In maintains only the raw unedited routes (NLRIs) that were received from the neighbors and thereby allows the inbound route policies to be processed again.

Enabling this feature can consume a significant amount of memory because the Adj-RIB-In table stays in memory. Inbound soft reconfiguration uses the address-family command `neighbor ip-address soft-reconfiguration inbound` for IOS nodes. IOS XR and NX-OS devices use the neighbor specific address-family command `soft-reconfiguration inbound`.

**IBGP**

The need for BGP within an AS typically occurs when the multiple routing policies exist, or when transit connectivity is provided between autonomous systems. In Figure 1-3, AS65200 provides transit connectivity to AS65100 and AS65300. AS65100 connects at R2, and AS65300 connects at R4.
Figure 1-3  AS65200 Provides Transit Connectivity

R2 could form a BGP session directly with R4, but R3 would not know where to route traffic from AS65100 or AS65300 when traffic from either AS reaches R3, as shown in Figure 1-4, because R3 would not have the appropriate route forwarding information for the destination traffic.

Figure 1-4  Transit Devices Need Full Routing Table

Adapting the full BGP table into an IGP is not a viable solution for the following reasons:

- **Scalability**: The Internet at the time of this writing has over 600,000 IPv4 networks and continues to increase in size. IGPs cannot scale to that level of routes.

- **Custom Routing**: Link state protocols and distance vector routing protocols use metric as the primary method for route selection. IGP protocols always use this routing pattern for path selection. BGP uses multiple steps to identify the best path and allows for BGP path attributes to manipulate the path for a specific prefix (NLRI). The path could be longer, which would normally be deemed suboptimal from an IGP protocol's perspective.
- **Path Attributes:** All the BGP path attributes cannot be maintained within IGP protocols. Only BGP is capable of maintaining the path attribute as the prefix is advertised from one edge of the AS to the other edge.

**IBGP Full Mesh Requirement**

It was explained earlier in this chapter how BGP uses the AS_PATH as a loop detection and prevention mechanism because the ASN is prepended when advertising to an EBGP neighbor. IBGP peers do not prepend their ASN to the AS_PATH, because the NLRIs would fail the validity check and would not install the prefix into the IP routing table.

No other method exists to detect loops with IBGP sessions, and RFC 4271 prohibits the advertisement of a NLRI received from an IBGP peer to another IBGP peer. RFC 4271 states that all BGP routers within a single AS must be fully meshed to provide a complete loop-free routing table and prevent traffic blackholing.

In Figure 1-5, R1, R2, and R3 are all within AS65100. R1 has an IBGP session with R2, and R2 has an IBGP session with R3. R1 advertises the 10.1.1.0/24 prefix to R2, which is processed and inserted into R2’s BGP table. R2 does not advertise the 10.1.1.0/24 NLRI to R3 because it received the prefix from an IBGP peer. To resolve this issue, R1 must form a multihop IBGP session so that R3 can receive the 10.1.1.0/24 prefix directly from R1. R1 connects to R3’s 10.1.23.3 IP address, and R3 connects to R1’s 10.1.12.1 IP address. R1 and R3 need a static route to the remote peering link, or R2 must advertise the 10.1.12.0/24 and 10.1.23.0/24 network into BGP.

---

**Figure 1-5  IBGP Prefix Advertisement Behavior**
Peering via Loopback Addresses

BGP sessions are sourced by the outbound interface toward the BGP peers IP address by default. Imagine three routers connected via a full mesh. In the event of a link failure on the R1-R3 link, R3's BGP session with R1 times out and terminates. R3 loses connectivity to R1's networks even though R1 and R3 could communicate through R2 (multihop path). The loss of connectivity occurs because IBGP does not advertise routes learned from another IBGP peer as in the previous section.

Two solutions exist to overcome the link failure:

- Add a second link between all routers (3 links will become 6 links) and establish two BGP sessions between each router.
- Configure an IGP protocol on the routers’ transit links, advertise loopback interfaces into the IGP, and then configure the BGP neighbors to establish a session to the remote router's loopback address.

Of the two methods, the second is more efficient and preferable.

The loopback interface is virtual and always stays up. In the event of link failure, the session remains intact while the IGP finds another path to the loopback address and, in essence, turns a single-hop IBGP session into a multihop IBGP session.

Updating the BGP configuration to set the destination of the BGP session to the remote router's loopback IP address is not enough. The source IP address of the BGP packets will still reflect the IP address of the outbound interface. When a BGP packet is received, the router correlates the source IP address of the packet to the BGP neighbor table. If the BGP packet source does not match an entry in the neighbor table, the packet cannot be associated to a neighbor and is discarded.

The source of BGP packets can be set statically to an interface's primary IP address with the BGP session configuration command `neighbor ip-address update-source interface-type interface-number` on IOS nodes. IOS XR and NX-OS devices use the command `update-source interface-type interface-number` under the neighbor session within the BGP router configuration.

Figure 1-6 illustrates the concept of peering using loopback addresses after the 10.1.13.0/24 network link fails. R1 and R3 still maintain BGP session connectivity while routes learned from OSPF allow BGP communication traffic between the loopbacks via R2. R1 can still forward packets to R3 via R2 because R1 performs a recursive lookup to identify R2 as the next-hop address.
EBGP

EBGP peerings are the core component of the BGP protocol on the Internet. EBGP is the exchange of network prefixes between autonomous systems. The following behaviors are different on EBGP sessions when compared to IBGP sessions:

- Time to Live (TTL) on BGP packets is set to one. BGP packets drop in transit if a multihop BGP session is attempted (TTL on IBGP packets is set to 255, which allows for multihop sessions).
- The advertising router modifies the BGP next-hop to the IP address sourcing the BGP connection.
- The advertising router prepends its ASN to the existing AS_PATH.
- The receiving router verifies that the AS_PATH does not contain an ASN that matches the local routers. BGP discards the NLRI if it fails the AS_PATH loop prevention check.

The configuration for EBGP and IBGP sessions are fundamentally the same on IOS, IOS XR, and NX-OS devices, except that the ASN in the `remote-as` statement is different from the ASN defined in the BGP process.
Note Different outbound (or inbound) route policies may be different from neighbor-to-neighbor, which allows for a dynamic routing-policy within an AS.

EBGP learned paths always have at least one ASN in the AS_PATH. If multiple ASNs are listed in the AS_PATH, the most recent AS is always prepended (the furthest to the left). The BGP attributes for all paths to a specific network prefix can be shown with the command show bgp ipv4 unicast network on IOS, IOS XR, and NX-OS devices.

Example 1-8 displays the BGP path attributes for the remote prefix (192.168.3.3/32).

**Example 1-8  BGP Prefix Attributes for Remote Prefix**

```
R1-IOS# show bgp ipv4 unicast 192.168.3.3
BGP routing table entry for 192.168.3.3/32, version 11
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  65200 65300
  10.1.12.2 from 10.1.12.2 (192.168.2.2)
  Origin IGP, localpref 100, valid, external, best
```

Table 1-5 explains the output provided in Example 1-8 and its correlation to BGP. Some of the BGP path attributes may change depending on the BGP features used.

**Table 1-5  BGP Prefix Attributes**

<table>
<thead>
<tr>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paths: (1 available, best #1)</td>
<td>Provides a count of BGP paths in the BGP Loc-RIB and identifies the path selected as the BGP best path. All the paths and BGP attributes are listed after this.</td>
</tr>
<tr>
<td>Not advertised to any peer</td>
<td>Identifies whether the prefix was advertised to a BGP peer or not. BGP neighbors are consolidated into BGP update-groups. Explicit neighbors can be seen with the command show bgp ipv4 unicast update-group on IOS or IOS XR nodes.</td>
</tr>
<tr>
<td>65200 65300</td>
<td>This is the AS_PATH for the NLRI as it was received.</td>
</tr>
</tbody>
</table>
### Output Description

<table>
<thead>
<tr>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.12.2 from 10.1.12.2 (192.168.2.2)</td>
<td>The first entry lists the IP address of the EBGP edge peer. The <em>from</em> field lists the IP address of the IBGP router that received this route from the EBGP edge peer. (In this case, the route was learned from an EBGP edge peer, so the address will be the EBGP edge peer.) Expect this field to change when an external route is learned from an IBGP peer. The number in parentheses is the BGP Identifier (RID) for that node.</td>
</tr>
<tr>
<td>Origin IGP</td>
<td>The Origin is the BGP well-known mandatory attribute that states the mechanism for advertising this route. In this instance, it is an Internal route.</td>
</tr>
<tr>
<td>metric 0</td>
<td>Displays the optional nontransitive BGP attribute <em>Multiple-Exit Discriminator (MED)</em>, also known as BGP metric.</td>
</tr>
<tr>
<td>localpref 100</td>
<td>Displays the well-known discretionary BGP attribute Local Preference.</td>
</tr>
<tr>
<td>valid</td>
<td>Displays the validity of this path.</td>
</tr>
<tr>
<td>External</td>
<td>Displays how the route was learned. It will be internal, external, or local.</td>
</tr>
</tbody>
</table>

### EBGP and IBGP Topologies

Combining EBGP sessions with IBGP sessions can cause confusion in terminology and concepts. Figure 1-6 provides a reference topology for clarification of concepts. R1 and R2 form an EBGP session, R3 and R4 form an EBGP session as well, and R2 and R3 form an IBGP session. R2 and R3 are IBGP peers and follow the rules of IBGP advertisement, even if the routes are learned from an EBGP peer.

As an EBGP prefix is advertised to an IBGP neighbor, issues may arise with the NLRI passing the validity check and the next-hop reachability check preventing advertisements to other BGP peers. The most common issue involves the failure of the next-hop accessibility. IBGP peers do not modify the next-hop address if the NLRI has a next-hop address other than 0.0.0.0. The next-hop address must be resolvable in the global RIB for it to be valid and advertised to other BGP peers.

To demonstrate this concept, only R1 and R4 have advertised their loopback interfaces into BGP, 192.168.1.1/32, and 192.168.4.4/32. Figure 1-7, displays the BGP table for all four routers. Notice that the BGP best path symbol (>) is missing for the 192.168.4.4/32 prefix on R2, and for the 192.168.1.1/32 on R3.
R1’s BGP table is missing the 192.168.4.4/32 prefix because the prefix did not pass R2’s next-hop accessibility check preventing the execution of the BGP best path algorithm. R4 advertised the prefix to R3 with the next-hop address of 10.1.34.4, and R3 advertised the prefix to R2 with a next-hop address of 10.1.34.4. R2 does not have a route for the 10.1.34.4 IP address and deems the next-hop inaccessible. The same logic applies to R1’s 192.168.1.1/32 prefix when advertised toward R4.

Example 1-9 shows the BGP attributes on R3 for the 192.168.1.1/32 prefix. Notice that the prefix is not advertised to any peer because the next-hop is inaccessible.

Example 1-9   BGP Path Attributes for 192.168.1.1/32

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>*192.168.1.1/32</td>
<td>0.0.0.0</td>
</tr>
</tbody>
</table>

To correct the issue, the peering links, 10.1.12.0/24 and 10.1.34.0/24, need to be in both R2’s and R3’s routing table via either technique:

- IGP advertisement. Remember to use the passive interface to prevent an accidental adjacency from forming. Most IGPs do not provide the filtering capability like BGP.
- Advertising the networks into BGP.

Both techniques allow the prefixes to pass the next-hop accessibility test.

Figure 1-8 displays the topology with both transit links advertised into BGP. Notice that this time all four prefixes are valid with a BGP best path selected.
Next-Hop Manipulation

Imagine a service provider network with 500 routers and every router containing 200 EBGP peering links. To ensure that the next-hop address is reachable to the IBGP peers requires the advertisement of 100,000 peering networks in BGP or an IGP consuming router resources.

Another technique to ensure that the next-hop address check passes without advertising peering networks into a routing protocol involves the modification of the next-hop address in the BGP advertisement. The next-hop IP address can be modified on inbound or outbound neighbor routing policies. Managing IP addresses in a route policy can be a complicated task. Configuring the `next-hop-self` address-family feature modifies the next-hop address in all external NLRIs using the IP address of the BGP neighbor.

The command `neighbor ip-address next-hop-self [all]` is used for each neighbor under the address-family configuration on IOS nodes, and the command `next-hop-self` is applied under the neighbor address-family configuration for IOS XR and NX-OS devices.

Figure 1-9 shows the topology and BGP routing table for all four routers. Notice that R2 and R3 advertised the EBGP routes to each other with the next-hop address as the BGP session IP address, allowing the NLRIs to pass the next-hop accessibility check.
IBGP Scalability

The inability for BGP to advertise a prefix learned from one IBGP peer to another IBGP peer can lead to scalability issues within an AS. The formula \( n(n-1)/2 \) provides the number of sessions required where \( n \) represents the number of routers. A full mesh topology of 5 routers requires 10 sessions, and a topology of 10 routers requires 45 sessions. IBGP scalability becomes an issue for large networks.

Route Reflectors

RFC 1966 introduces the concept that an IBGP peering can be configured so that it reflects routes to another IBGP peer. The router reflecting routes is known as a route reflector (RR), and the router receiving reflected routes is a route reflector client. Three basic rules involve route reflectors and route reflection:

**Rule #1:** If a RR receives a NLRI from a non-RR client, the RR advertises the NLRI to a RR client. It does not advertise the NLRI to a non-route-reflector client.

**Rule #2:** If a RR receives a NLRI from a RR client, it advertises the NLRI to RR client(s) and non-RR client(s). Even the RR client that sent the advertisement...
receives a copy of the route, but it discards the NLRI because it sees itself as
the route originator.

**Rule #3:** If a RR receives a route from an EBGP peer, it advertises the route to RR
client(s) and non-RR client(s).

Figure 1-10 demonstrates the route reflector rules.
Only route reflectors are aware of this change in behavior because no additional BGP configuration is performed on route-reflector clients. BGP route reflection is specific to each address-family. The command `neighbor ip-address route-reflector-client` is used on IOS nodes, and the command `route-reflector-client` is used on IOS XR and NX-OS devices under the neighbor address-family configuration.

### Loop Prevention in Route Reflectors

Removing the full mesh requirements in an IBGP topology introduces the potential for routing loops. When RFC 1966 was drafted, two other BGP route reflector specific attributes were added to prevent loops.

**ORIGINATOR_ID**, an optional nontransitive BGP attribute is created by the first route reflector and sets the value to the RID of the router that injected/advertised the route into the AS. If the ORIGINATOR ID is already populated on an NLRI, it should not be overwritten.

If a router receives a NLRI with its RID in the Originator attribute, the NLRI is discarded.

**CLUSTER_LIST**, a nontransitive BGP attribute, is updated by the route reflector. This attribute is appended (not overwritten) by the route reflector with its cluster-id. By default this is the BGP identifier. The cluster-id can be set with the BGP configuration command `bgp cluster-id cluster-id` on IOS and IOS XR nodes. NX-OS devices use the command `cluster-id cluster-id`.

If a route reflector receives a NLRI with its cluster-id in the Cluster List attribute, the NLRI is discarded.

Example 1-10 provides sample output prefix output from a route that was reflected. Notice that the originator ID is the advertising router and that the cluster list contains two route-reflector IDs listed in the order of the last route reflector that advertised the route.

### Example 1-10  Route Reflector Originator ID and Cluster List Attributes

```plaintext
RP/0/0/CPU0:R1-XR# show bgp ipv4 unicast 10.4.4.0/24
# Output omitted for brevity
Paths: (1 available, best #1)
  Local 10.1.34.4 from 10.1.12.2 (192.168.4.4)
    Origin IGP, metric 0, localpref 100, valid, internal, best, group-best
    Received Path ID 0, Local Path ID 1, version 7
    Originator: 192.168.4.4, Cluster list: 192.168.2.2, 192.168.3.3
```

### Out-of-Band Route Reflectors

As explained earlier, BGP can establish multihop BGP sessions and does not change the next-hop path attribute when routes are advertised to IBGP neighbors. Some large network topologies use dedicated BGP routers for route reflection that are outside of the data path.
These out-of-band route reflectors provide control plane programming for the BGP routers that are in the data path and only require sufficient memory and processing power for the BGP routing table. Out-of-band route reflectors should not use the `next-hop-self`, or it will place the route reflector into the data path. Organizations that use MPLS L2VPNs, L3VPNs, and so on will use multiple out-of-band route reflectors for exchanging BGP path information.

**Confederations**

RFC 3065 introduced the concept of BGP confederations as an alternative solution to IBGP full mesh scalability issues shown earlier. A confederation consists of sub-ASs known as a Member-AS that combine into a larger AS known as an AS Confederation. Member ASs normally use ASNs from the private ASN range (64512-65535). EBGP peers from the confederation have no knowledge that they are peering with a confederation, and they reference the confederation identifier in their configuration.

Figure 1-11 demonstrates a BGP confederation with the confederation identifier of AS200. The Member-ASs are AS65100 and AS65200. R3 provides route reflection in Member-AS 65100.

Confederations share behaviors from both IBGP sessions and EBGP sessions. The changes are as follows:

- The AS_PATH attribute contains a subfield called AS_CONFED_SEQUENCE. The AS_CONFED_SEQUENCE is displayed in parentheses before any external ASNs in the AS_PATH. As the route passes from Member-AS to Member-AS, the AS_CONFED_SEQUENCE is appended to contain the Member-AS ASNs.
AS_CONFED_SEQUENCE attribute is used to prevent loops, but it is not used (counted) when choosing shortest AS_PATH.

- Route reflectors can be used within the Member-AS like normal IBGP peerings.
- The BGP MED attribute is transitive to all other Member-ASs, but does not leave the confederation.
- The LOCAL_PREF attribute is transitive to all other Member-ASs, but does not leave the confederation.
- IOS XR nodes do not require a route policy when peering with a different Member-AS, even though the remote-as is different.
- The next-hop address for external confederation routes does not change as the route is exchanged between Member-AS to Member-AS.
- The AS_CONFED_SEQUENCE is removed from the AS_PATH when the route is advertised outside of the confederation.

Configuring a BGP confederation is shown in the following steps:

**Step 1.** Create the BGP Routing Process. Initialize the BGP process with the global command `router bgp member-asn`.

**Step 2.** Set the BGP Confederation Identifier. Identify the BGP confederations with the command `bgp confederation identifier as-number`. The `as-number` is the BGP confederation ASN.

**Step 3.** Identify Peer Member-ASs. On routers that directly peer with another Member-AS, identify the peering Member-AS with the command `bgp confederation peers member-asn`.

**Step 4.** Configure BGP confederation members as normal; the remaining configuration follows normal BGP configuration guidelines.

Example 1-11 displays R1’s and R2’s BGP table. R1 resides in AS100 and does not see any of the BGP subconfederation information. R1 is not aware the AS200 is subdivided into a BGP confederation.

R2’s BGP table participates in the Member-AS 65100. Notice the next-hop address is not modified for the 10.67.1.0/24 (Network between R6 and R7) even though a Member-AS. The AS_CONFED_SEQUENCE is listed in parentheses to indicate it passed through Sub-AS 65200 in the AS200 confederation.
Example 1-11  RI’s and R2’s BGP Table

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.12.0/24</td>
<td>10.1.12.2</td>
<td>0</td>
<td>0</td>
<td>200</td>
<td>i</td>
</tr>
<tr>
<td>10.1.23.0/24</td>
<td>10.1.23.2</td>
<td>0</td>
<td>0</td>
<td>200</td>
<td>i</td>
</tr>
<tr>
<td>10.1.25.0/24</td>
<td>10.1.25.2</td>
<td>0</td>
<td>0</td>
<td>200</td>
<td>i</td>
</tr>
<tr>
<td>10.1.34.0/24</td>
<td>10.1.34.2</td>
<td>0</td>
<td>0</td>
<td>200</td>
<td>i</td>
</tr>
<tr>
<td>10.1.46.0/24</td>
<td>10.1.46.2</td>
<td>0</td>
<td>0</td>
<td>200</td>
<td>i</td>
</tr>
<tr>
<td>10.1.56.0/24</td>
<td>10.1.56.2</td>
<td>0</td>
<td>0</td>
<td>200</td>
<td>i</td>
</tr>
<tr>
<td>10.1.67.0/24</td>
<td>10.1.67.2</td>
<td>0</td>
<td>0</td>
<td>200</td>
<td>i</td>
</tr>
</tbody>
</table>

Example 1-12 displays the NLRI information for 10.67.1.0/24 from the perspective of R2. Notice that the NLRI from within a confederation includes the option of confed-internal and confed-external for sources.

Example 1-12  Confederation NLRI

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.12.0/24</td>
<td>0.0.0.0</td>
<td>32768</td>
<td>0</td>
<td></td>
<td>i</td>
</tr>
<tr>
<td>10.1.23.0/24</td>
<td>10.23.3</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>i</td>
</tr>
<tr>
<td>10.1.25.0/24</td>
<td>10.1.25.5</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>[65200] i</td>
</tr>
<tr>
<td>10.1.34.0/24</td>
<td>10.1.34.4</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>[65200] i</td>
</tr>
<tr>
<td>10.1.46.0/24</td>
<td>10.1.46.6</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>[65200] i</td>
</tr>
<tr>
<td>10.1.56.0/24</td>
<td>10.1.56.6</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>[65200] i</td>
</tr>
<tr>
<td>10.1.67.0/24</td>
<td>10.1.67.6</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>[65200] i</td>
</tr>
</tbody>
</table>

BGP routing table entry for 10.1.67.0/24, version 8
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
    1 3
  Refresh Epoch 1
    (65200)
    10.56.1.6 from 10.1.25.5 (10.1.56.5)
    Origin IGP, metric 0, localpref 100, valid, confed-external
    rx pathid: 0, tx pathid: 0
BGP Communities

BGP communities provide additional capability for tagging routes and for modifying BGP routing policy on upstream and downstream routers. BGP communities can be appended, removed, or modified selectively on each attribute as the route travels from router to router.

*BGP communities* are an optional transitive BGP attribute that can traverse from *autonomous system* to *autonomous system*. A BGP community is a 32-bit number that can be included with a route. A BGP community can be displayed as a full 32-bit number (0-4,294,967,295) or as two 16-bit numbers (0-65535):(0-65535) commonly referred to as new-format.

*Private BGP communities* follow the convention that the first 16-bits represent the AS of the community origination, and the second 16-bits represent a pattern defined by the originating AS. The private BGP community pattern could vary from organization to organization, do not need to be registered, and could signify geographic locations for one AS while signifying a method of route advertisement in another AS. Some organizations publish their private BGP community patterns on websites, such as http://wwwONESC.net/communities/.

In 2006, RFC 4360 expanded BGP communities’ capabilities by providing an extended format. *Extended BGP communities* provide structure for various classes of information and are commonly used for VPN Services.

IOS XR and NX-OS devices display BGP communities in new-format by default, and IOS nodes display communities in decimal format by default. IOS nodes can display communities in new-format with the global configuration command `ip bgp-community new-format`.

Example 1-13 displays the BGP community in decimal format on top, and in new-format on bottom.
BGP Community Formats

<table>
<thead>
<tr>
<th>DECIMAL FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>R3# show bgp 192.168.1.1</td>
</tr>
<tr>
<td>Output omitted for brevity</td>
</tr>
<tr>
<td>BGP routing table entry for 192.168.1.1/32, version 6</td>
</tr>
<tr>
<td>Community: 6553602 6577023</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>New-Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>R3# show bgp 192.168.1.1</td>
</tr>
<tr>
<td>Output omitted for brevity</td>
</tr>
<tr>
<td>BGP routing table entry for 192.168.1.1/32, version 6</td>
</tr>
<tr>
<td>Community: 100:2 100:23423</td>
</tr>
</tbody>
</table>

IOS and NX-OS devices do not advertise BGP communities to peers by default. Communities are enabled on a neighbor-by-neighbor basis with the BGP address-family configuration command neighbor ip-address send-community [standard | extended | both], and NX-OS devices use the command send-community [standard | extended | both] under the neighbor address-family configuration. Standard communities are sent by default, unless the optional extended or both keywords are used.

IOS XR advertises BGP communities to IBGP peers by default, but EBGP peers require the neighbor address-family configuration command send-community-ebgp for advertising standard BGP communities, and the command send-extended-community-ebgp to advertise extended BGP communities. Both commands are required if both community formats are to be sent to an EBGP peer.

Route Summarization

Summarizing prefixes conserves router resource(s) and accelerates best path calculation by reducing the size of the table. Summarization also provides the benefit(s) of stability by reducing routing churn by hiding route flaps from downstream routers. Although most ISPs do not accept prefixes larger than /24 for IPv4 (/25-/32), the Internet, at the time of this writing, still has more than 600,000 routes and continues to grow toward a million routes. Route summarization is required to reduce the size of the BGP table for Internet routers.

BGP route summarization on EBGP routers for nontransitive ASs reduce route computation on routers in the core of the nontransitive AS. In Figure 1-12, R3 summarizes all the EBGP routes received from AS65100 and AS65200 to reduce route computation on R4 during link flaps. In the event of a link flap on the 10.1.13.0/24 network, R3 removes all AS65100 routes learned directly from R1 and identifies the same networks via R2 with a different (longer AS_PATH). R4 processes the same changes that R3 processes and is a waste of CPU cycles because R4 receives connectivity only from R3. If R3 summarized the network range, instead of running the best-path algorithm against multiple routes, the best-path algorithm would execute only once.
The two techniques for BGP summarization are the following:

- **Static**: Create a static route to Null 0 for the prefix, and then advertise the network via a network statement. The downfall to this technique is that the summary route will always be advertised even if the networks are not available.

- **Dynamic**: Configure an aggregation network range. When viable routes that match the network range enter the BGP table, an aggregate route is created. On the originating router, the aggregated prefix sets the next-hop to Null 0. The route to Null 0 is automatically created by BGP as a loop-prevention mechanism.

In both methods of route aggregation, a new network prefix with a shorter prefix length is advertised into BGP. Because the aggregated prefix is a new route, the summarizing router is the originator for the new aggregate route.

### Aggregate-Address

Dynamic route summarization is accomplished with the BGP address-family configuration commands identified in Table 1-6.

<table>
<thead>
<tr>
<th>OS</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOS</td>
<td><code>aggregate-address</code> <code>network</code> <code>subnet-mask</code> <code>summary-only</code> <code>suppress-map</code> <code>route-map-name</code> <code>as-set</code> <code>advertise-map</code> <code>route-map-name</code></td>
</tr>
<tr>
<td>IOS XR</td>
<td><code>aggregate-address</code> <code>network/prefix-length</code> <code>summary-only</code> <code>route-policy</code> <code>route-policy-name</code> <code>as-set</code> <code>advertise-map</code> <code>route-policy-name</code></td>
</tr>
<tr>
<td>NX-OS</td>
<td><code>aggregate-address</code> <code>[network subnet-mask network/prefix-length]</code> <code>summary-only</code> <code>suppress-map</code> <code>route-map-name</code> <code>as-set</code> <code>advertise-map</code> <code>route-map-name</code></td>
</tr>
</tbody>
</table>
The aggregate-address command advertises the aggregated route in addition to the original networks. Using the optional no-summary keyword suppresses the networks in the summarized network range. BGP considers aggregated addresses as local routes.

**Note** Aggregate addresses are local BGP routes when modifying BGP AD.

**Flexible Route Suppression**
Some traffic engineering designs require “leaking” routes, which is the advertisement of a subset of more specific routes in addition to performing the summary. Leaking routes can be done at the process by explicitly stating the prefixes to suppress, or on a neighbor level by indicating which prefixes should not be suppressed.

**Selective Prefix Suppression**
Selective prefix suppression explicitly lists the networks that should not be advertised along with the summary route to neighbor routers.

IOS and NX-OS uses a suppress-map, which uses the keyword suppress-map route-map-name instead of using the no-summary keyword. In the referenced route-map, only the prefixes that should be suppressed are permitted. IOS XR routers use the keyword route-policy route-policy-name in lieu of the no-summary keyword. In the route policy, the action command suppress is used after conditionally matching the prefixes that should be suppressed.

**Leaking Suppressed Routes**
The summary-only keyword suppresses all the more specific routes of an aggregate address from being advertised. After a route is suppressed, it is still possible to advertise the suppressed route to a specific neighbor.

IOS devices use an unsuppress-map with the BGP neighbor address-family configuration command neighbor ip-address unsuppress-map route-map-name. In the referenced route-map, only the prefixes that should be leaked are permitted. IOS XR routers use an outbound route policy with the action command unsuppress to indicate which prefixes should be leaked.

**Atomic Aggregate**
Aggregated routes act like new BGP routes with a shorter prefix length. When a BGP router summarizes a route, it does not advertise the AS path information from before the aggregation. BGP path attributes such as AS-Path, MED, and BGP communities are not included in the new BGP advertisement. The Atomic Aggregate attribute indicates that a loss of path information has occurred.
For example:

- R1 and R2 are advertising the 172.16.10/24 and 172.16.20/24 networks.
- R3 is aggregating the routes into the 172.16.0.0/22 network range, which is advertised to all of R3's peers.

Example 1-14 displays R3's BGP table. R1's BGP prefix 172.16.10/24 advertised to R3. Notice the AS-Path of 65100 and BGP Community of 100:100.

**Example 1-14**  
172.16.10/24 BGP Path Information

```
R3-IOS# show bgp ipv4 unicast 172.16.1.0
BGP routing table entry for 172.16.1.0/24, version 13
Paths: (1 available, best #1, table default, Advertisements suppressed by an aggregate.)
Not advertised to any peer
Refresh Epoch 1
65100
10.1.13.1 from 10.1.13.1 (192.168.1.1)
  Origin IGP, metric 0, localpref 100, valid, external, best
Community: 100:100
```

R3’s aggregate route (summary) does not include the BGP communities (including AS-Path history) for the routes in the summarization range. R3 advertises the aggregate route to R1 and R2, and those routers install the 172.16.0.0/22 summary route because their AS-Path is not listed in the AS-Path attribute and passes the AS-Path loop check.

Example 1-15 displays the BGP path information for the 172.16.0.0/22 summary network on R1. The AS-Path of the aggregated route displays only the aggregating router, but does not include the AS-Path of the routes being summarized (AS65100 or AS65200), nor is the BGP community included in the routes being summarized. The BGP path information indicates that this is an aggregated prefix and was aggregated by R3 (192.168.3.3). The Atomic-Aggregate in the route indicates a loss of information occurred during aggregation on the aggregating router.

**Example 1-15**  
172.16.0.0/22 BGP Path Information

```
R1-IOS# show bgp ipv4 unicast 172.16.0.0
BGP routing table entry for 172.16.0.0/21, version 5
Paths: (1 available, best #1, table default)
Not advertised to any peer
Refresh Epoch 1
100, (aggregated by 300 192.168.3.3)
10.1.13.3 from 10.1.13.3 (192.168.3.3)
  Origin IGP, metric 0, localpref 100, valid, external, atomic-aggregate, best
```
Route Aggregation with AS_SET

To keep the BGP path information history, the optional as-set keyword may be used with the aggregate-address command. As the router generates the aggregate route, BGP attributes from the summarized routes are copied over to it. The AS-Path settings from the original prefixes are stored in the AS_SET portion of the AS-Path. (The AS_SET is displayed within brackets, and counts only as one hop, even if multiple ASs are listed.)

Route Aggregation with Selective Advertisement of AS-SET

Using the AS-SET feature with network aggregation combines all the attributes of the original prefixes into the aggregated prefixes. This might cause issues with your routing policy. For example, if one of the prefixes contains the No-Export BGP community, the aggregate address will not be exported. To resolve these types of problems, selectively choose the routes that the path attributes will copy to the aggregate route. The use of the advertise-map option allows for conditionally matching and denying attributes that should be permitted or denied in the aggregated route.

Default Route Advertisement

Advertising a default route into the BGP table requires the default route to exist in the RIB and the BGP configuration command default-information originate to be used. The redistribution of a default route or use of a network 0.0.0.0/0 does not work without the default-information originate command.

Default Route Advertisement per Neighbor

Some network topologies restrict the size of the BGP advertisements to a neighbor because the remote router does not have enough processing power or memory for the full BGP routing table. Connectivity is still required, so the peering routers only advertise the default route to the remote router.

A default route is advertised to a BGP peer with the BGP address-family configuration command neighbor ip-address default-originate for IOS nodes or with the BGP neighbor address-family configuration command default-originate for IOS XR and NX-OS devices. Default route advertisement to a specific neighbor does not require a default route to be present in the RIB or BGP Loc-RIB table.

Note  A behavior difference between IOS and IOS XR occurs when a default route is already present in the BGP table. IOS nodes advertise the route as if it was the originating router. (None of the existing attributes are passed to the peer.) IOS XR nodes advertise the network to the peer as it exists in the BGP table with the entire default route attributes (AS-Path, and so on).
Remove Private AS

Some organizations might not be able to meet the qualifications for obtaining their own ASN but still want to receive Internet routing tables from their service provider. In these situations, the service provider may assign the organization a private ASN for peering. Private ASNs should not be advertised by the service provider to other ISPs on the Internet.

The feature `remove private AS` removes the private AS of routes that are advertised to the configured peer. The router performs the following path analysis with the `remove private AS` feature:

- Removes only private ASNs on routes advertised to EBGP peers.
- If the AS-Path for the route has only private ASNs, the private ASNs are removed.
- If the AS-Path for the route has a private ASN between public ASNs, it is assumed that this is a design choice, and the private ASN is not removed.
- If the AS-Path contains confederations (AS_CONFED_SEQ), BGP removes the private AS numbers only if they are included after the AS_CONFED_SEQ (Confederation AS-Path) of the path.

The `remove private AS` feature is configured on IOS nodes with the BGP address-family configuration command `neighbor ip-address remove-private-as`. IOS XR and NX-OS devices use the BGP neighbor address-family configuration command `remove-private-as`.

Allow AS

The `Allow AS` feature allows for routes to be received and processed even if the router detects its own ASN in the AS-Path. A router discards BGP network prefixes if it sees its ASN in the AS-Path as a loop prevention mechanism. Some network designs use a transit AS to provide connectivity to two different locations. BGP detects the network advertisements from the remote site as a loop and discards the route. The AS-Path loop check feature needs to be disabled to maintain connectivity in scenarios such as these.

On IOS nodes, the command `neighbor ip-address allowas-in` is placed under the address-family. IOS XR and NX-OS nodes use the BGP neighbor address-family configuration command `allowas-in`.

LocalAS

When two companies merge, one of the ASNs is usually returned to the regional Internet registry (RIR). During the migration, each company needs to maintain its own ASN while changes are made with its peering neighbors to update their configuration.
The LocalAS feature is configured on a per peer basis, and allows for BGP sessions to establish using an alternate ASN than the ASN that the BGP process is running on. The LocalAS feature works only with EBGP peerings.

IOS nodes use the BGP address-family neighbor configuration command `neighbor ip-address local-as alternate-as-number [no-prepend [replace-as [dual-as]]].` IOS XR and NX-OS devices use the equivalent command `local-as alternate-as-number [no-prepend [replace-as [dual-as]]]` under the neighbor. By default, the alternate ASN is added to the AS-Path for routes that are sent and received between these two peers.

One problem with the alternate ASN being prepended when receiving the routes is that other IBGP peers drop the network prefixes as part of a routing loop detection.

- To stop the alternate ASN from being prepended when receiving routes, the optional keyword `no-prepend` is used.
- To stop the alternate ASN from being prepended when sending routes, the optional keywords `no-prepend replace-as` is used.
- If both `no-prepend replace-as` keywords are used, all routers see the BGP advertisements as if they were running the original AS in the BGP process.

After the remote peer changes the remote-as setting on the BGP configuration, the `local-as` commands should be removed. If the coordination of maintenance windows cannot occur during the same time, the `no-prepend replace-as dual-as` optional keywords allow the remote peer to user either ASN for the BGP session. The remote BGP router peers with the ASN in the router process statement, or the alternate ASN in the `local-as` configuration.

**Summary**

BGP is a powerful path vector routing protocol that provides scalability and flexibility that cannot be compared to any other routing protocol. BGP uses TCP port 179 for all BGP communication between peers, which allows BGP to establish sessions with directly attached routers or with routers that are multiple hops away.

Originally, BGP was intended for the routing of IPv4 prefixes between organizations, but over the years has had significant increase in functionality and feature enhancements. BGP has expanded from being an Internet routing protocol to other aspects of the network, including the data center.

BGP provides a scalable control-plane signaling for overlay topologies, including MPLS VPNs, IPsec SAs, and VXLAN. These overlays can provide Layer 3 services, such as L3VPNs, or Layer 2 services, such as eVPNs, across a widely used scalable control plane for everything from provider-based services to data center overlays. Every AFI / SAFI combination maintains an independent BGP table and routing policy, which makes BGP the perfect control plane application.
This chapter provided a fundamental overview of BGP from a session perspective, as well as route advertisement behaviors for IPv4 and IPv6 protocols. Networking vendors continue to use BGP for new features, and having the ability to effectively troubleshoot BGP is becoming more and more necessary.

This book provides emphasis on various BGP-related problems that are encountered in real-life deployments, which have caused major outages to the network over the years.

References


Index

Numbers

6PE (IPv6 provider edge routers), 607–611
configuration, 611–615
verification and troubleshooting, 615–620
6VPE (IPv6 VPN provider edge), 620–622
configuration, 627–629
control plane verification, 629–633
data plane verification, 633–638
IPv6-aware VRF, 622–623
next-hop, 623–627

A

AC (attachment circuit), 545
ACL-based traffic mirroring, 61–62
ACLs (access control lists)
AS-Path ACLs, 188–190
checking in path, 91
filtering prefixes, 174–175
filtering sessions, 429–431
verifying packet reception, 90
Active state, 10
AD (administrative distance), 5
additional-paths selection command, 732
add-path feature, 726–738
address families, 3–4
address-family ipv6 labeled-unicast command, 612
address-family l2vpn evpn command, 778
address-family link-state link-state command, 763
address-family vpnv4 unicast command, 262
Adj-RIB-in table, 17
Adj-RIB-out table, 17
advertisement interval, 226, 243–244
advertisement-interval command, 226
advertising
default routes, 42, 508
between PE and CE routers, 487
af-group command, 295
AFI (address-family identifier), 3–4
aggregate-address command, 39–40, 42
aggregate-address summary-only command, 149
aggregation. See route summarization
AIGP (Accumulated Interior Gateway Protocol), 381–383
aigp command, 381–382
aigp send med command, 383
allocate-label command, 612
Allow AS feature, 43
allowas-in command, 43
ALTO (Application Layer Traffic Optimization), 756–757
ARP suppression, 655–656
ASNs (autonomous system numbers), 2
LocalAS feature, 43–44
removing private ASNs, 43
as-path length command, 319
as-path-loopcheck out disable command, 164
AS-Path, 3, 162–164
ACLs, 188–190
length in best path calculation, 383
maximum length, 318–322
relax feature, 377
troubleshooting L3VPN, 509–513
AS_SET attribute, 42
asterisk (*) query modifier, 184–185
asymmetric IRB, 657
asynchronous mode (BFD), 713–715
asynchronous mode with echo function (BFD), 715
Atomic Aggregate attribute, 40–41
attestations, 441–442
attributes, tuning memory consumption, 284–293
authentication, 424–427
Authorization Certificates, 443
autodiscovery bgp command, 571
autodiscovery bgp signaling bgp command, 582
autodiscovery in VPLS, 569–579
AS (autonomous system), 2, 43–44
bad network design, troubleshooting, 160–162
best path calculation, 20–21, 379–389, 417
AIGP (Accumulated Interior Gateway Protocol), 381–383
AS-Path length, 383
cluster list, 388
computing and installing new path, 226–227
EBGP (external BGP), 386
IBGP (internal BGP), 386
IGP (Interior Gateway Protocol), 386–387
local origination, 380
local preference, 380
MED (Multi-Exit Discriminator), 384–386
neighbor addresses, 388–389
oldest path, 387
Origin attribute, 383–384
router-id, 387
for routing table, 394–395
for RPKI, 460–463
troubleshooting, 389–390
visualizing topology, 390–394
weight, 380
best-external feature, 738–741
bestpath med confed command, 384
bestpath med missing-as-worst command, 385
bestpath med non-deterministic command, 386
bestpath origin-as allow invalid command, 462
BFD (bidirectional fast detection), 218
BFD (bidirectional forwarding detection), 712–713
asynchronous mode, 713–715
asynchronous mode with echo function, 715
configuration and verification, 715–724
troubleshooting, 724–726
bfd command, 715
bfd echo command, 722
bfd fast-detect command, 715
bf interval min rx multiplier command, 715
BGP (Border Gateway Protocol), 1–2
add-path feature, 726–738
best-external feature, 738–741
BGP FRR and PIC, 741–753
configuration component requirements, 11
on IOS XR routers, 12–13
for MPLS L3VPN, 497–502
on NX-OS routers, 13–14
dynamic BGP peering, 138–142
fast-external-fallover feature, 726
GR (Graceful-Restart) feature, 693–700
IGP (Interior Gateway Protocol) versus, 758–759
IPv6 BGP, 591–611
missing network prefixes, 185–203
new features. See new features
NSR (nonstop routing), 700–712
peer flapping issues. See peer flapping issues, troubleshooting
peering down issues. See peering down issues, troubleshooting
redistribution into IGP, 413–416
route advertisement issues. See route advertisement issues, troubleshooting
route convergence explained, 205–207
troubleshooting, 216–227
route flapping, troubleshooting, 246–250
scaling functions, 288–322
impact of growing Internet routing tables, 283–285
Internet routing tables on Cisco platforms, 285–288
route reflectors, 322–364
BGP-LS (BGP for Link-State Distribution) 791

BGP flowspec, 467–479

importance of, 419–420

interdomain routing, 431–463

RTBH filtering, 463–466

sessions, 420–431

slow peers, 237–246

update generation, 212–216

update groups, 207–212

verification for MPLS L3VPN, 502–506

VxLAN EVPN, 653–690

bgp additional-paths command, 729, 739

bgp additional-paths install command, 732, 733, 746, 748, 752, 753

bgp additional-paths select backup command, 355

bgp additional-paths select best command, 737

bgp additional-paths select command, 732

bgp advertise-best-external command, 739, 748, 753

bgp always-compare-med command, 294, 385

bgp best path igp-metric ignore command, 387

bgp bestpath as-path multipath-relax command, 377

bgp bestpath compare-routerid command, 387

bgp bestpath igp-metric ignore command, 350

bgp bestpath med always command, 385

bgp bestpath med confed command, 384

bgp bestpath med missing-as-worst command, 385

bgp bestpath origin-as allow invalid command, 461

bgp bestpath origin-as use origin-as validity command, 461

bgp bestpath origin-as use validity command, 461

bgp bestpath prefix-validate allow-invalid command, 461

bgp bestpath prefix-validate disable command, 459

bgp cluster-id command, 33, 324, 327

bgp deterministic-med command, 294, 386

bgp fast-external-fallover command, 726

bgp fast-external-fallover disable command, 726

BGP flowspec, 467–479

BGP for tunnel setup, 771–773

bgp graceful-restart command, 696

bgp graceful-restart purge-time command, 696

bgp graceful-restart restart-time command, 696

bgp graceful-restart stalepath-time command, 696

bgp graceful-restart stalepath-timer command, 696

bgp import-delay command, 262

BGP I/O process, 256–258

bgp label-delay command, 262

bgp listen command, 142

bgp maxas-limit command, 319, 321

bgp maximum neighbor command, 322

bgp nexthop route-map command, 225

bgp nexthop trigger delay command, 224–225

BGP NHT feature, 223–225

selective tracking, 225–226

bgp origin-as validation disable command, 459

bgp origin-as validation signal ibgp command, 458

bgp origin-as validation time command, 460

bgp recursion host command, 753

bgp redistribute-internal command, 152

bgp refresh command, 306–307

bgp refresh max-eor-time command, 306–307

bgp refresh stalepath-time command, 306–307

BGP Router process, 255–256

bgp router-id command, 7, 13, 500

BGP Router-ID (RID), 7

gbg rpki server tcp port refresh command, 449

BGP Scanner process, 219–222, 253–255

bgp scan-time command, 222

BGP signaling

in VPLS, 580–586

in VPWS, 558–560

bgp slow-peer detection command, 245

bgp slow-peer split-update-group dynamic configuration command, 246

bgp sso route-refresh-enable command, 702

BGP tables

fields in, 19–20

network prefix and path attributes, 17–20

for route advertisement, 152–154

BGP-LS (BGP for Link-State Distribution), 757–759

BGP-LS NLRI, 759–761

configuration, 762–771

Path attribute, 762
BGP-LS NLRI, 759–761
BGP-PA (BGP Policy Accounting), 604–607
bgp-policy accounting command, 605–606
blocked processes in IOS XR, 103–106
brackets ([[]] query modifier, 181–182
buffered logging, 75–76

C

cache size, verifying, 241
capturing traffic. See sniffing
caret (^) query modifier, 180–181
caret in brackets ([^]) query modifier, 182
CE routers
default route advertisement, 508
network advertisement, 487
cef table output-chain build favor convergence-speed command, 745
Cisco VIRL, 51
clear bgp command, 317
clear bgp graceful command, 22
clear bgp ipv4 unicast * soft in command, 312–313
clear bgp ipv4 unicast in command, 301
clear bgp ipv4 unicast soft command, 299
clear bgp ipv4 unicast update-group command, 209
clear bgp out command, 300
clear bgp slow command, 246
clear bgp soft command, 22
clear cef interface bgp-policy-statistics command, 607
clear cef interface policy-statistics command, 607
clear ip bgp in command, 300
clear ip bgp soft command, 22
clear ip bgp soft in command, 305
clear tcp pcb command, 95
clear tcp tcb command, 95, 257
cluster list in best path calculation, 388
Cluster-ID, 165–167
collector-id command, 33, 324
communications, 5–6
communities, 37–38, 167–173, 185
conditional BGP debugs, 199–203
conditional matching, 174
ACLs (access control lists), 174–175
of BGP communities, 185
prefix matching, 175–177
confederations, 34–37
configuration
6PE, 611–615
6VPE, 627–629
BFD, 715–724
BGP
component requirements, 11
explicitly configured peers, 421–424
on IOS routers, 11–12
on IOS XR routers, 12–13
for MPLS L3VPN, 497–502
on NX-OS routers, 13–14
verifying for peering down issues, 84–87
BGP flowspec, 469–479
BGP signaling in VPWS, 560
BGP-LS, 762–771
confederations, 35
dynamic BGP peering, 139–142
EBGP and IBGP multipath configuration, 370–372
EIBGP multipath configuration, 372–377
L3VPN (Layer3 VPN), 487–488
lab devices, 52–56
ORF, 312–316
PBB-EVPN, 778–787
RPKI, 449–460
VPLS, 562–564
VPWS, 550–558
VxLAN EVPN, 661–690
VxLAN flood-and-learn, 647–652
Connect state, 9
connectivity. See reachability of peers
cone logging, 75
core plane (6VPE), 624–626
verification, 629–633
core words, 547
convergence. See route convergence
CoPP (Control Plane Policing), 127–138
copp profile strict command, 133
CPU issues
high utilization, 251–267
in peer flapping, 125–127
tuning, 295–308
cross-link, peering on, 402–411

D

data plane (6VPE), 626–627
verification, 633–638
debg bgp command, 307–308
debg bgp ipv4 unicast command, 110–111
debug bgp ipv4 unicast events command, 110–111
disable-peer-as-check command, 164
distribute bgp-ls command, 763
distributed anycast gateway, 654–655
diverse path, 346–349
documentation, importance of, 48
dollar sign ($) query modifier, 181
drop threshold command, 133
dynamic BGP peering, troubleshooting, 138–139
configuration, 142
dynamic refresh update groups, 302–305
dynamic route summarization, 39
aggregate-address command, 39–40
dynamic slow peers, 245–246
documentation, importance of, 48
dollar sign ($) query modifier, 181
drop threshold command, 133
E

EBGP (external BGP), 5, 26–28
in best path calculation, 386
confederations versus, 34–35
mandatory route policy for IOS XR, 172–173
multihop, 427–429
multipath configuration, 370–372
next-hop manipulation, 30–31
topologies, 28–30
ebgp-multihop 2 command, 427
ebgp-multihop command, 92, 427
ECMP (equal cost multipath), 21
dynamic BGP peering, troubleshooting, best path calculation, 377–390
full mesh with IBGP, 412
multihoming and multipath, 367–377
peering on cross-link, 402–411
race conditions, 397–402
redistribution into IGP, 413–416
transit routing, 395–397
visualizing topology, 390–394
EEM (Embedded Event Manager), 57
EGP (Exterior Gateway Protocol), 1
EIBGP multipath configuration, 372–377
encapsulation, BGP tunneling, 771–773
distributed anycast gateway, 654–655
enhanced route refresh, 305–308
event-history command, 108
events
tracing, 77–80
triggering in lab, 56–57
EVPN (Ethernet VPN)
PBB-EVPN, 773–787
VxLAN EVPN, 653–690
explicitly configured peers, 421–424
extended BGP communities, 37
extended community-based ORF, 309–310
extended EVPN communities, 777

default route advertisement, 42, 222–223, 508
default information originate command, 42
default-metric command, 385
default-originate command, 42
direct sessions, multihop sessions versus, 5–6
debg bgp packets command, 102
debg bgp policy-execution events command, 197
debg bgp route-server command, 364
debg bgp update command, 200, 215, 307–308
deb g bgp updates command, 199
debg bgp vpnv4 unicast addpath command, 751
debg ip bgp command, 110–111
debg ip bgp update command, 215
debg ip tcp transaction command, 111
debg logfile bgp command, 201
debg logfile command, 77
debug sockets tcp command, 111
debg tcp packet command, 111
debgs
conditional BGP debugs, 199–203
for peering issues, 110–112
decoding messages, 99–103
default route advertisement, 42, 222–223, 508
default information originate command, 42
default metric command, 385
default originate command, 42
direct sessions, multihop sessions versus, 5–6
GR (Graceful-Restart) feature, 693–700
graceful-restart-helper command, 700

HARD RESETS, 22
hardware access-list team region arp-ether 256 command, 666
hardware requirements for lab setup, 51
hierarchical route reflectors, 331–332
high availability
BFD (bidirectional forwarding detection), 712–726
BGP add-path feature, 726–738
BGP best-external feature, 738–741
BGP fast-external fallover feature, 726
BGP FRR and PIC, 741–753
BGP GR (Graceful-Restart) feature, 693–700
BGP NSR (nonstop routing), 700–712
high CPU issues
in peer flapping, 125–127
troubleshooting, 251–252
capturing CPU history, 265
on IOS, 252–258
on IOS XR, 258–262
on NX-OS, 262–264
sporadic high CPU conditions, 265–267
Hold Time attribute, 6
hold timer expired, 116–119
hold-queue in command, 117
hw-module bfd-hw-offload enable location command, 721–722
hyphen (-) query modifier, 182

F
fabric forwarding anycast-gateway-mac command, 655
fabric forwarding mode
anycast-gateway command, 655
failure detection, 218–227
fast-external-fallover feature, 726
feature bfd command, 715
feature bgp command, 85
feature mpls l3vpn command, 496
feature mpls ldp command, 496
feature-set mpls command, 496
filter vlan command, 63
filtering
ORF (Outbound Route Filtering), 309–316
prefixes, 173–185
RTBH filtering, 463–466
for session security, 429–431
firewalls, checking in path, 91
flapping. See peer flapping issues, troubleshooting; route flapping
flexible route suppression, 40
flood-and-learn mechanism, 645–653
flowspec (BGP), 467–479
FSM (Finite State Machine), 8
full mesh requirement in IBGP, 24
full mesh with IBGP, troubleshooting multihoming, 412

G
gateways
distributed anycast gateway, 654–655
VxLAN gateway types, 645
generic, 547
Internet routing tables
scaling on Cisco platforms, 285–288
size impact of, 283–285
tuning memory consumption, 290–292
inter-router communication, 5–6
interworking, 549–550
IOS
AS-Path ACLs, 188–190
BGP basic configuration, 11–12
BGP configuration for MPLS
L3VPN, 497–498
conditional BGP debugs, 200
CoPP configuration, 128
Error-Subcode values, 99
high CPU issues, 252–258
memory consumption, 269–274
peer templates, 297–298
peer-groups, 295
prefix lists, 186–188
RID allocation in, 7
route-maps, 192–196
SPAN on, 58–59
VRF creation, 488–489
IOS XR
BGP basic configuration, 12–13
BGP configuration for MPLS
L3VPN, 499–500
BGP templates, 295–296
blocked processes, troubleshooting, 103–106
conditional BGP debugs, 200–201
decoding BGP messages, 101–102
high CPU issues, 258–262
LPTS on, 134–138
mandatory EBGP route policy, 172–173
memory consumption, 274–277
RID allocation in, 7
route convergence, 227–234
RPL (route policy language), 196–198
SPAN on, 60–62
tracing in, 106–108
TCP on, 55
VRF creation, 489–490
ip access-group command, 91
ip access-list command, 430
ip bgp fast-external-fallover command, 726
ip bgp-community new-format command, 37
ip cef command, 126
ip flowspec disable command, 473
ip tcp path-mtu-discovery command, 121
ip verify unicast source reachable-via command, 466
ip vrf command, 489, 622
ip vrf forwarding command, 489, 627
Iperf, 52
IPsec (Internet Protocol Security), 431, 439
ipv4 bgp policy accounting command, 605–606
ipv4 flowspec disable command, 473
IPv4 peering, IPv6 reachability over, 596–601
IPv4 routes over IPv6 next-hop, 601–604
ipv6 access-group command, 91
ipv6 address link-local command, 421
IPv6 BGP
6PE over MPLS, 607–620
6VPE, 620–638
BGP-PA (BGP Policy Accounting), 604–607
IPv4 over IPv6 next-hop, 601–604
next-hop, 591–596
peering with link-local addresses, 421–424
reachability over IPv4 peering, 596–601
ipv6 bgp policy accounting command, 605–606
ipv6 flowspec disable command, 473
ipv6 link-local command, 421
IPv6 peers, troubleshooting, 112–113
ipv6 traffic-filter command, 91
IPv6-aware VRF, 622–623
IRB (integrated route/bridge) modes, 656–658
J-L
Jumbo MTU, 219
KEEPALIVE message, 7
L2VPN (Layer2 VPN), 482
services, 543–545
terminology, 545–547
VPLS (Virtual Private LAN Service), 561–588
VPWS (Virtual Private Wire Service), 548–560
L3VPN (Layer3 VPN), 482, 483
BGP configuration, 497–502
BGP verification, 502–506
collection, 487–488
MP-BGP (Multi-Protocol BGP), 486
network advertisement, 487
RD (route distinguisher), 485
RT (route target), 485–486
RT constraints, 534–538
services, 524–534
troubleshooting, 506–524
VRF (Virtual Routing and Forwarding), 483–485
VRF creation, 488–491
VRF verification, 492–495
lab
configuring lab devices, 52–56
setting up, 49–51
triggering events, 56–57
label exchange, 538–540
Layer 3 traffic mirroring, 60–61
leaking routes, 40
link-local addresses, 421–424
link-state distribution, 755–759
BGP-LS NLRI, 759–761
BGP-LS Path attribute, 762
configuration, 762–771
local origination in best path calculation, 380
local preference in best path calculation, 380
local route advertisement, troubleshooting, 145–147
local-as command, 44
Local-AS community, 170–171
LocalAS feature, 43–44
local-install interface-all command, 472
local-preference command, 380
Loc-RIB table, 17
logging, 74–77
logging host vrf command, 77
logging hostnameprefix command, 77
longest match path selection, 377–379
Looking Glass, 185
loop prevention, 3
in IBGP, 24
in route reflectors, 33
loopback addresses
in IBGP, 25–26
loopback-to-loopback ping testing, 87–88
LPTS (Local Packet Transport Services), 134–138
missing prefixes, troubleshooting, 185–186
conditional BGP debugs, 199–203
incomplete configuration of routing policies, 198–199
AS-Path ACLs, 188–190
prefix lists, 186–188
route-maps, 191–196
RPL (route policy language), 196–198
missing routes, troubleshooting, 156–157
bad network design, 160–162
BGP communities, 167–173
conditionally matching BGP communities, 185
filtering prefixes by routing policy, 173–185
next-hop check failures, 157–160
validity check failure, 162–167
mls rate-limit command, 127
monitor session command, 58
monitor session session-id filter command, 59
MP-BGP (Multi-Protocol BGP), 3–4, 486, 658–661
MPLS (Multiprotocol Label Switching), 481–483
6PE over, 607–620
6VPE over, 620–638
BGP configuration, 497–502
BGP verification, 502–506
forwarding, 495–496, 541–542
L2VPN (Layer2 VPN), 543–588
L3VPN (Layer3 VPN). See L3VPN
label exchange, 538–540
mpls ip command, 496
mpls ldp command, 496
neighbor aigp send med
command, 383
neighbor allowas-in command, 43
neighbor announce rpki state
command, 458
neighbor as-override
command, 512
neighbor default-originate
command, 42, 292
neighbor disable-connected-
check command, 86, 427
neighbor dont-capability-
negotiate enhanced-refresh
command, 308
neighbor ebgp-multihop
command, 86, 92, 427, 428
neighbor fall-over command, 218
neighbor graceful-restart
command, 699
neighbor graceful-restart-
helper command, 700
neighbor ha-mode graceful-
restart command, 699
neighbor ha-mode sso
command, 702
neighbor ip-address fall-over
bfd command, 715
neighbor local-as command,
44
neighbor local-preference
command, 380
neighbor maximum-prefix
command, 316
neighbor maximum-refix
command, 317
neighbor next-hop-self
command, 30
neighbor prefix-length-size
command, 573
neighbor remote-as command,
422
neighbor remove-private-as
command, 43
neighbor route-reflector-client
command, 33
neighbor route-server-client
command, 360, 362
neighbor send-community
command, 38, 196, 294, 458
neighbor slow-peer-split-
update-group static
command, 245
neighbor soft-reconfiguration
inbound command, 22,
273, 299
neighbor transport single-
session command, 115
neighbor ttl-security
command, 86
neighbor ttl-security hops
command, 428
neighbor unsuppress-map
command, 40
neighbor update-source
command, 25, 86
neighbor weight command,
380
neighbor-group command,
295
neighbors, limiting number
of, 322
Netdr capture, 66–67
network advertisement. See
advertising
network mask route-map
command, 17
network prefix in BGP tables,
17–20
network route-map command,
17
network route-policy
command, 17
network statements, 17
new features
BGP for tunnel setup,
771–773
link-state distribution,
755–771
PBB-EVPN, 773–787
next-hop
in 6VPE, 623–627
in IPv6 BGP, 591–596
selective tracking, 225–226
tracking, 223–225
next-hop check failures, troubleshooting, 157–160
next-hop manipulation, 30–31
next-hop route-policy command, 225
next-hop trigger-delay command, 224–225
next-hop trigger-delay critical command, 224–225
next-hop-self command, 30–31, 159, 342, 412
NLRI (Network Layer Reachability Information), 3
BGP-LS NLRI, 759–761
EVPN NLRI and routes, 776–777
no bgp client-to-client reflection, 323
no bgp client-to-client reflection command, 327, 330
no bgp client-to-client reflection intra-cluster cluster-id command, 330
no bgp default ip4-unicast command, 11
no bgp enforce-first-as command, 361
no bgp fast-external-fallover command, 726
no bgp next-hop trigger enable command, 224
no bgp recursion host command, 753
no echo disable command, 722
no ip redirects command, 715
no ip route-cache cef command, 126
no next-hop resolution prefix-length minimum 32 command, 753
no shut command, 650
no shutdown command, 62
No Advertise community, 167–168
No Export community, 169–170
No Export SubConfed community, 170–171
no-summary command, 40
NOTIFICATION message, 8
notifications, Error code and Error-Subcode values, 96–99
NSR (nonstop routing), 700–712
nsr command, 702
nsr process-failures switchover command, 704
nv overlay command, 660–661
NX-OS
AS-Path ACLs, 188–190
BGP basic configuration, 13–14
BGP configuration for MPLS L3VPN, 500–502
conditional BGP debugs, 201–203
CoPP on, 129–134
decoding BGP messages, 102–103
Ethanalyzer, 70–74
high CPU issues, 262–264
memory consumption, 278–281
peer templates, 296–297
prefix lists, 186–188
RID allocation in, 7
route convergence, 234–236
route-maps, 192–196
SPAN on, 62–63
tracing in, 108–110
VRF creation, 490–491
oldest path in best path calculation, 387
OPEN message, 6–7
OpenConfirm state, 10
OpenSent state, 10
option additional-paths install command, 733
ORF (Outbound Route Filtering), 309
configuration, 312–316
extended community-based ORF, 309–310
format, 310–312
prefix-based ORF, 309
Origin AS validation, 443–445
ROA, 445
RPKI best path calculation, 460–463
RPKI configuration and verification, 449–460
RPKI prefix validation, 446–448
Origin attribute in best path calculation, 383–384
Originator-ID, 165–167
outbound policy, changing, 242–243
out-of-band route reflectors, 33–34
OutQ value, verifying, 240
overlay routing, 481
on VxLAN, 645
BGP EVPN, 653–690
flood-and-learn mechanism, 645–653
as-override command, 512–513
PA (path attributes), 3
in best path calculation, 20–21
in BGP tables, 17–20
packets. See also messages determining loss location and direction, 88–89
sniffing, 57–58
with EPC tool, 68–70
P
with Ethalyzer, 70–74
on IOS routers, 58–59
on IOS XR routers, 60–62
with Netdr capture, 66–67
on NX-OS routers, 62–63
platform-specific tools, 65
with RSPAN, 63–64
tunneling, 771–773
verifying transmittal, 89–90
verifying with ACLs, 90
VxLAN packet structure, 643–644
parentheses and pipe (|) query modifier, 183
partitioned route reflectors, 332–339
pass through (BGP authentication), 426–427
Path attribute (BGP-LS), 762
paths
add-path feature, 726–738
best path calculation. See best path calculation computing and installing, 226–227
diverse path, 346–349
loop prevention, 3
multihoming and multipath, 367–377
PA (path attributes), 3
route filtering, 21–22
tuning memory consumption, 292–293
pbb edge i-sid core-bridge command, 778
PCE (Path Computation Elements), 756–757
PE node failure, 752
PE routers
default route advertisement, 508
network advertisement, 487
PE-CE link failure, 748–752
peer flapping issues, troubleshooting, 115
bad BGP updates, 115–116
CoPP (Control Plane Policing), 127–138
high CPU issues, 125–127
hold timer expired, 116–119
MTU mismatch issues, 120–124
peer status, 8–10
peer templates
on IOS, 297–298
on NX-OS, 296–297
peer-groups, 295
peering down issues, troubleshooting, 83–84
BGP debugging, 110–112
BGP message decoding, 99–103
BGP notifications, 96–99
BGP traces in IOS XR, 106–108
BGP traces in NX-OS, 108–110
blocked processes in IOS XR, 103–106
IPv6 peers, 112–113
single session versus multisession case study, 113–115
verifying configuration, 84–87
verifying reachability, 87–96
peers
dynamic BGP peering, 138–142
explicitly configuring, 421–424
IPv6 reachability, 596–601
peering on cross-link, 402–411
slow peers, 237–246
update generation, 212–216
update groups, 207–212
period (.) query modifier, 183
periodic BGP scan, 219–222
permit ip any any any command, 90
PIC (Prefix Independent Convergence), 741–742
BGP PIC core feature, 742–745
BGP PIC edge feature, 745–753
ping mpls ipv4 command, 541, 564
ping testing, 87–90
ping vrf command, 495
PKI (Public Key Infrastructure), 439–441
platform rate-limit command, 127
plus sign (+) query modifier, 183–184
PMTUD (Path-MTU Discovery), 120–124
Policy Certificates, 443
prefix attributes, 27–28
prefix hijacking, 432–439
prefix lists, 186–188
prefix matching, 175–177
prefix suppression, 40
prefix-based ORF, 309
prefixes
filtering, 173–185
maximum prefixes, 316–318
troubleshooting missing prefixes. See missing prefix, troubleshooting
tuning memory consumption, 290
prefix-length-size 2 command, 573
private ASNs, removing, 43
private BGP communities, 37
problems
identifying, 48
reproducing, 49
configuring lab devices, 52–56
setting up lab for, 49–51
triggers
triggering events in lab, 56–57
understanding, 48–49
process restart command, 106, 281
processes
blocked processes in IOS XR, 103–106
restarting, 106, 281
PW (pseudowires), 546–547
Q
query modifiers (regular expressions), 178–185
question mark (?) query modifier, 184
R
race conditions, 397–402
RD (route distinguisher), 485
rd auto command, 573
reachability of peers
IPv6 over IPv4, 596–601
verifying, 87–96
receiving routes, 154–155
recursion host, 752–753
redistribution, BGP into IGP, 413–416
refresh-time command, 449
regular expressions, filtering prefixes, 177–185
remote-as command, 26
Remove Private AS feature, 43
remove-private-as command, 43
reproducing problems, 49
configuring lab devices, 52–56
setting up lab for, 49–51
resiliency in service providers, 370
restart bgp command, 281
restarting processes, 106, 281
ROA (Route Origination Authorization), 445
route advertisement issues, troubleshooting
aggregation, 147–149
bad network design, 160–162
BGP communities, 167–173
BGP tables, 152–154
conditionally matching BGP communities, 185
filtering prefixes by routing policy, 173–185
local issues, 145–147
missing routes, 156–157
next-hop check failures, 157–160
receiving and viewing routes, 154–155
redistribution, 150–152
validity check failure, 162–167
route convergence explained, 205–207
troubleshooting, 216–217
failure detection, 218–227
on IOS XR, 227–234
on NX-OS, 234–236
route filtering, 21–22
route flapping, troubleshooting, 246–250
route leaking, 524
route policies
filtering prefixes by, 173–185
mandatory EBGP route policy for IOS XR, 172–173
troubleshooting, 185–203
route redistribution, troubleshooting, 150–152
route reflectors, 31–33
loop prevention, 33
out-of-band route reflectors, 33–34
scaling with, 322–364
route refresh
enhanced route refresh, 305–308
soft reconfiguration versus, 298–302
Route Servers, 185
route servers, 357–364
route summarization, 38–39
AS_SET attribute, 42
aggregate-address command, 39–40
Atomic Aggregate attribute, 40–41
flexible route suppression, 40
troubleshooting, 147–149
Routed mode (firewalls), 92
route-map command, 191, 604
route-maps, 191–196
route-policy command, 40, 604
router bgp command, 255–256
route-reflector-client command, 33
router-id command, 7
router-id in best path calculation, 387
routing protocols
BGP, 1–2
IGP versus EGP, 1
RPKI
best path calculation, 460–463
configuration and verification, 449–460
prefix validation, 446–448
rpki server transport tcp port command, 449
RPL (route policy language), 196–198
RSPAN (Remote SPAN), 63–64
RT (route target), 485–486
6VPE next-hop, 624
constraints, 534–538
troubleshooting, 520–524
RTBH (remote triggered black-hole) filtering, 463–466
run show_processes -m -h -t command, 275

S

SAFI (subsequent address-family identifier), 3–4
S-BGP (Secure BGP), 439–442
scalability of IBGP, 31
scaling BGP (Border Gateway Protocol)
functions, 288–322
impact of growing Internet routing tables, 283–285
Internet routing tables on Cisco platforms, 285–288
route reflectors, 322–364
securing BGP (Border Gateway Protocol)
BGP flowspec, 467–479
importance of, 419–420
interdomain routing, 431–463
RTBH filtering, 463–466
sessions, 420–431
SECURITY message, 443
selective next-hop tracking, 225–226
selective prefix suppression, 40
selective route download, 339–342
send-community command, 38
send-community-ebgp command, 38
send-extended-community-ebgp command, 38
service instance ethernet command, 553
service password-encryption command, 425
service provider resiliency, 370
service timestamps command, 76
service-policy input command, 127–128
services
L2VPN, 543–545
L3VPN, 524–534
session-group command, 295
sessions
direct versus multihop, 5–6
peer status states, 8–10
resets, 298–302
securing, 420–431
shadow sessions, 355–357
simulating, 95–96
TCP sessions, verifying, 94–95
types of, 4–5
verification, 14–17
set local-preference command, 380
set origin command, 384
set traffic-index command, 604
set weight command, 380
shadow route reflectors, 349–355
shadow sessions, 355–357
show bfd counters packet private detail location command, 724
show bfd neighbors command, 718
show bfd neighbors details command, 721
show bfd neighbors hardware details command, 721
show bfd session command, 718
show bgp all safi command, 706
show bgp bestpath command, 389
show bgp cluster-ids command, 330
show bgp command, 18, 158, 190, 234, 250, 454, 456
show bgp community command, 167
show bgp community local-as command, 171
show bgp community no-advertise command, 168
show bgp convergence detail vrf all command, 235
show bgp event-history command, 109
show bgp event-history periodic command, 110–111
show bgp flowspec summary command, 471, 473
show bgp internal mem-stats detail command, 279
show bgp ipv4 flowspec summary command, 471, 473
show bgp ipv4 rt-filter command, 538
show bgp ipv4 unicast 192.168.11 command, 356
show bgp ipv4 unicast cluster-ids internal command, 330
show bgp ipv4 unicast command, 27, 454
show bgp ipv4 unicast neighbor advertised-routes command, 351
show bgp ipv4 unicast neighbor command, 113, 240, 705
show bgp ipv4 unicast neighbors advertised-routes command, 740
show bgp ipv4 unicast neighbors command, 696, 702
show bgp ipv4 unicast regex 300 command, 180
show bgp ipv4 unicast regex 100 command, 179
show bgp ipv4 unicast replication command, 214
show bgp ipv4 unicast summary command, 141, 208, 240
show bgp ipv4 unicast summary slow command, 246
show bgp ipv4 unicast update-group command, 208
show bgp ipv4 unicast update-group performance-statistics command, 233
show bgp ipv4 unicast vrf command, 518
show bgp ipv6 command, 596
show bgp ipv6 labeled-unicast neighbors command, 615
show bgp ipv6 summary command, 615
show bgp ipv6 unicast command, 594, 617
show bgp ipv6 unicast neighbors command, 615
show bgp ipv6 unicast summary command, 615
show bgp l2vpn evpn command, 667, 675–676, 780
show bgp l2vpn evpn summary command, 667, 780
show bgp l2vpn evpn vni-id command, 667
show bgp l2vpn vpls command, 585
show bgp l2vpn vpls summary command, 574
show bgp link-state link-state command, 766, 770
show bgp link-state link-state summary command, 766
show bgp neighbor command, 300, 702, 729
show bgp neighbors command, 15, 696
show bgp nsr command, 706
show bgp origin-as validity command, 454, 456
show bgp origin-as validity invalid command, 455
show bgp origin-as validity not-found command, 455
show bgp origin-as validity valid command, 455
show bgp paths command, 289
show bgp process command, 702
show bgp regexp command, 177
show bgp route-server context command, 363
show bgp rpki server command, 450
show bgp rpki servers command, 450
show bgp rpki summary command, 450, 460, 461
show bgp rpki table command, 452
show bgp rfilter unicast command, 538
show bgp sessions command, 707
show bgp summary command, 14, 119, 271
show bgp summary nsr command, 706
show bgp summary nsr standby command, 706
show bgp trace command, 107–108
show bgp trace error command, 108
show bgp trace sync command, 710–711
show bgp unicast command, 502, 504
show bgp update in error neighbor detail command, 101
show bgp update-group command, 210
show bgp vpv4 unicast all replication command, 241
show bgp vpv4 unicast all summary command, 240
show bgp vpv4 unicast convergence command, 233
show bgp vpv4 unicast rd command, 519, 520
show bgp vpv6 unicast all summary command, 630
show bgp vpv6 unicast rd command, 632
show bgp vpv6 unicast summary command, 630, 632
show bgp vpv6 unicast vrf labels command, 632
show bgp vrf ABC all neighbors received prefix-filter command, 314
show bgp vrf all all summary command, 264
show bgp vrf command, 504
show bgp vrf vpv6 unicast command, 629
show cef interface bgp-policy-statistics command, 606
show cef interface policy-statistics command, 606
show cef vrf ipv6 hardware command, 634
show clock command, 247–248
show debug logfile command, 77, 201
show evpn evi command, 786
show evpn evi detail command, 786
show flowspec client command, 475–478
show flowspec client internal command, 478
show flowspec nlri command, 473
show forwarding ipv6 route command, 637
show forwarding route command, 235
show hardware rate-limit command, 127
show ibc l in rate command, 67
show interface accounting command, 636
show interface command, 89–90, 117, 606
show interface nv1 command, 650
show ip bgp attr nexthop command, 224
show ip bgp replication command, 241
show ip bgp summary command, 15, 247–248
show ip cef vrf command, 749
show ip interface brief command, 493
show ip interface brief vrf all command, 493
show ip interface command, 89–90, 492
show ip route bgp command, 234, 340
show ip route command, 159, 248
show ip route repair-paths command, 751
show ip route summary command, 255
show ip route vrf* all command, 248
show ip spd command, 117
show ip traffic command, 88–89
show ipv4 traffic command, 89
show ipv4 vrf all interface brief command, 493
show ipv6 cef ipv6-address command, 618
show ipv6 route vrf command, 629
show l2route evpn evi command, 670
show l2route evpn fl all command, 686
show l2route evpn imet evi command, 686
show l2vpn atom vc command, 565
show l2vpn atom vc detail command, 555
show l2vpn bridge-domain autodiscovery bgp command, 576
show l2vpn bridge-domain bd-name command, 576
show l2vpn bridge-domain command, 565, 781
show l2vpn bridge-domain detail command, 781
show l2vpn bridge-domain summary command, 564
show l2vpn discovery bridge-domain command, 575
show l2vpn forwarding bridge-domain mac-address command, 785–786
show l2vpn internal event-history command, 586
show l2vpn internal event-trace command, 586
show l2vpn pbb backbone-source-mac command, 785–786
show l2vpn service vfi name command, 576
show l2vpn signaling rib command, 584
show l2vpn signaling rib detail command, 584
show l2vpn trace command, 586
show l2vpn vfi name command, 564, 575
show l2vpn xconnect detail command, 555
show logging command, 276
show lpts ifib all brief command, 136
show lpts pfib brief command, 137
show lpts pfib hardware entry brief command, 135
show lpts pfib hardware police command, 135
show mac address-table vlan command, 652
show memory compare command, 276, 277
show memory compare end command, 277
show memory compare report command, 277
show memory compare start command, 277
show memory debug leaks command, 270
show memory statistics command, 270
show memory summary detail command, 276
show mls cef exception status command, 269
show mls cef maximum-routes command, 269
show monitor capture buffer command, 69
show monitor session command, 59
show monitor-session command, 60
show mpls forwarding command, 619, 636, 787
show mpls forwarding labels hardware command, 636
show mpls forwarding vrf command, 632
show mpls l2transport vc command, 555
show mpls l2transport vc vcid command, 565
show mpls ldp neighbor command, 553
show mpls switching command, 637
show nve interface command, 650
show nve internal event-history event command, 686
show nve internal platform interface command, 651
show nve internal platform interface nve command, 671
show nve peers command, 651, 668, 677
show nve peers detail command, 668
show nve vni command, 652, 686
show nve vni detail command, 652
show parser command, 107
show policy-map control-plane command, 128
show policy-map interface control-plane command, 132
show process bgp command, 258
show process blocked command, 105
show process command, 104
show process cpu command, 252, 254
show process cpu details command, 264
show process cpu sorted command, 125, 253
show process memory command, 271
show process threadname command, 260
show processes command, 254–255, 275
show processes cpu command, 258
show processes cpu history command, 125, 265
show processes cpu sort command, 262
show processes memory command, 275, 276, 279
show processes memory sorted command, 270–271
show processes threadname command, 260
show redundancy command, 705
show route command, 751
show routing unicast event-history add-route command, 264
show run rpl command, 196
show running-config command, 131–132
show snmp command, 125
show sockets internal event-history events command, 109–110
show system internal forwarding adjacency command, 637
show system internal forwarding vrf ipv6 route command, 637
show system internal memory-alerts-log command, 278
show system internal processes cpu command, 263
show system internal processes-name mem-stats detail command, 279
show system resources command, 278
show tcp brief all command, 141
show tcp brief command, 9, 257, 708
show tcp dump-file command, 710
show tcp dump-file list command, 710
show tcp nr brief command, 708
show tcp nr detail pcb command, 709
show tcp nr session-set brief command, 708
show tcp packet-set brief command, 709
show tech packet-trace command, 709
show tech netstack command, 110
show tech-platform l2vpn platform command, 588
show tech-support bgp command, 588, 712
show tech-support l2vpn command, 588
show tech-support routing bgp command, 588
show tech-support tcp nr command, 712
show vlan internal usage command, 66
show vrf command, 492
show vrf interface command, 492
show watchdog threshold memory command, 275
show xconnect all command, 565
shutdown command, 281
signaling
in VPLS, 580–586
in VPWS, 558–560
signaling disable command, 582
simulating sessions, 95–96
single session versus multisession case study, 113–115
slow peers, 237–238
detection of, 239–241
mitigation of, 242–246
show commands, 246
symptoms of, 238–239
SndWnd, verifying, 240–241
sniffing, 57–58
  with EPC tool, 68–70
  with Ethanalyzer, 70–74
  on IOS routers, 58–59
  on IOS XR routers, 60–62
  with Netdr capture, 66–67
  on NX-OS routers, 62–63
  with platform-specific tools, 65
  with RSPAN, 63–64
soBGP (Secure Origin BGP), 442–443
soft reconfiguration, route refresh versus, 298–302
soft resets, 22
soft-reconfiguration inbound command, 22, 302
software requirements for lab setup, 51
SPAN (Switched Port Analyzer)
  on IOS routers, 58–59
  on IOS XR routers, 60–62
  on NX-OS routers, 62–63
  RSPAN, 63–64
spd enable command, 117
spd headroom command, 117
S-PE (switching PE), 545
sporadic high CPU conditions, 265–267
static route summarization, 39
static slow peers, 245
suboptimal routing, troubleshooting, 514–520
summary fields, 15
summary-only command, 40
suppress-map command, 40
suppress-signaling-protocol ldp command, 582
symmetric IRB, 658
syslog logging, 76–77

triggers of problems
  triggering events in lab, 56–57
  understanding, 48–49
troubleshooting
  6PE, 615–620
  best path calculation, 389–390
  BFD (bidirectional forwarding detection), 724–726
dynamic BGP peering, 138–142
  edge architectures. See edge architectures, troubleshooting
  high CPU issues, 251–267
  L3VPN (Layer3 VPN), 506–524
  memory consumption, 267–281
  multihoming, 395–416
  peer flapping issues. See peer flapping issues, troubleshooting
  peering down issues. See peering down issues, troubleshooting
  route advertisement issues. See route advertisement issues, troubleshooting
  route convergence, 216–236
  route flapping, 246–250
  route policies, 185–203
  VPLS (Virtual Private LAN Service), 586–588
troubleshooting methodologies
  event tracing, 77–80
  identifying problem, 47–48
  logging, 74–77
  packet sniffers. See packets, sniffing
  reproducing problem, 49–56
  triggering events, 56–57
  understanding variables/triggers, 48–49
TTCP (Test TCP) utility, 52–56
TTL security, 428–429
ttl-security command, 428
tuning
CPU, 295–308
memory consumption, 284–290
tunneling packets, 771–773. See also VPNs (virtual private networks)

U
underlay networks, 481
underscore (_) query modifier, 179–180
unsuppress command, 40
update generation, 212–216
update groups, 207–212
UPDATE message, 7
update-source command, 25, 422

V
validation, Origin AS, 443–445
ROA, 445
RPKI best path calculation, 460–463
RPKI configuration and verification, 449–460
RPKI prefix validation, 446–448
validity check failure, troubleshooting, 162–167
variables, problem triggers
triggering events in lab, 56–57
understanding, 48–49
VC labels, 547
verification
6PE, 615–620
6VPE control plane, 629–633
6VPE data plane, 633–638
BFD, 715–724
BGP and BPM process state, 104–105
BGP for MPLS L3VPN, 502–506
blocked processes, 105
configuration for peering issues, 84–87
OutQ value, 240
PBB-EVPN, 778–787
reachability for peering issues, 87–96
route convergence, 227–234
RPKI, 449–460
sessions, 14–17
SnDWnd, 240–241
VPLS, 564–569
VPWS, 550–558
VRF (Virtual Routing and Forwarding), 492–495
VxLAN EVVPN, 661–690
VxLAN flood-and-learn, 647–652
viewing routes, 154–155
VIRL, 51
virtual route reflectors, 342–346
vn-segment-vlan-based command, 660–661
VPLS (Virtual Private LAN Service), 544, 561–588
autodiscovery, 569–579
BGP signaling, 580–586
configuration, 562–564
troubleshooting, 586–588
verification, 564–569
VPNs (virtual private networks), 481
6VPE. See 6VPE
MPLS. See MPLS (Multiprotocol Label Switching)
VPNv4 RRs (route reflectors), suboptimal routing with, 514–520
VPWS (Virtual Private Wire Service), 544, 548–560
BGP signaling, 558–560
configuration and verification, 550–558
interworking, 549–550
VRF (Virtual Routing and Forwarding), 483–485
creating, 488–491
IPv6-aware VRF, 622–623
verification, 492–495
vrf definition command, 489, 622, 627
vrf forwarding command, 489, 627
vrf upgrade-cli multi-af-mode command, 489
vrf upgrade-cli multi-af-mode vrf command, 623
VxLAN (Virtual Extensible LAN), 641–643
BGP EVVPN, 653–690
gateway types, 645
overlay, 645–653
packet structure, 643–644

W–Z
weight command, 380
weight in best path calculation, 380
xconnect group command, 560