VYATTA, INC. | Vyatta System

BGP

REFERENCE GUIDE

Vyatta
Suite 200
1301 Shoreway Road
Belmont, CA 94002
vyatta.com
650 413 7200
1 888 VYATTA 1 (US and Canada)

COPYRIGHT

Vyatta reserves the right to make changes to software, hardware, and documentation without notice. For the most recent version of documentation, visit the Vyatta web site at vyatta.com.

PROPRIETARY NOTICES

ISSUE DATE: February 2009

DOCUMENT REVISION. VC5 v03

RELEASED WITH: VC5.0.2

PART NO. A0-0116-10-0002

Table of Contents

Quick Reference to Commands

Use this section to help you quickly locate a command.

Quick List of Examples

Use this list to help you locate examples you’d like to try or look at.

Preface

This guide explains how to deploy the Border Gateway Protocol (BGP) on the Vyatta system. It describes the available commands and provides configuration examples.

This preface provides information about using this guide. The following topics are covered:

- Intended Audience
- Organization of This Guide
- Document Conventions
- Vyatta Publications

Intended Audience

This guide is intended for experienced system and network administrators. Depending on the functionality to be used, readers should have specific knowledge in the following areas:

- Networking and data communications
- TCP/IP protocols
- General router configuration
- Routing protocols
- Network administration
- Network security

Organization of This Guide

This guide has the following aid to help you find the information you are looking for:

- **Quick Reference to Commands**

  Use this section to help you quickly locate a command.

- **Quick List of Examples**

  Use this list to help you locate examples you'd like to try or look at.

This guide has the following chapters:

Chapter	Description	Page
Chapter 2: Global and Router-Specific Configuration	This chapter describes commands for global and router-specific configuration for BGP.	72
Chapter 3: Route Reflection	This chapter describes commands for for BGP route reflection.	155
Chapter 4: Confederations	This chapter describes commands for BGP confederations.	164
Chapter 5: Neighbors and Peer Groups	This chapter describes commands for BGP neighbors and peer groups.	170
Chapter 6: Route Redistribution	This chapter describes commands for BGP route redistribution.	271
Chapter 7: Route Server	This chapter describes commands for BGP route server.	283

Document Conventions

This guide contains advisory paragraphs and uses typographic conventions.

Advisory Paragraphs

This guide uses the following advisory paragraphs:

Warnings alert you to situations that may pose a threat to personal safety, as in the following example:

WARNING *Risk of injury. Switch off power at the main breaker before attempting to connect the remote cable to the service power at the utility box.*

Cautions alert you to situations that might cause harm to your system or damage to equipment, or that may affect service, as in the following example:

CAUTION *Risk of loss of service. Restarting a running system will interrupt service.*

Notes provide information you might need to avoid problems or configuration errors:

NOTE *You must create and configure network interfaces before enabling them for routing protocols.*

Typographic Conventions

This document uses the following typographic conventions:

`Courier`	Examples, command-line output, and representations of configuration nodes.
`boldface Courier`	In an example, your input: something you type at a command line.
boldface	In-line commands, keywords, and file names .
italics	Arguments and variables, where you supply a value.
<key>	A key on your keyboard. Combinations of keys are joined by plus signs (“+”). An example is <Ctrl>+<Alt>+<Del>.

[*arg1* | *arg2*]	Enumerated options for completing a syntax. An example is [enable | disable].
num1–numN	A inclusive range of numbers. An example is 1–65535, which means 1 through 65535.
arg1..*argN*	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, and eth3.
arg [*arg* ...] *arg*,[*arg*,...]	A value that can optionally represent a list of elements (a space-separated list in the first case, and a comma-separated list in the second case).

Vyatta Publications

More information about the Vyatta system is available in the Vyatta technical library, and on www.vyatta.com and www.vyatta.org.

Full product documentation is provided in the Vyatta technical library. To see what documentation is available for your release, see the *Guide to Vyatta Software Documentation*. This guide is posted with every release of Vyatta software and provides a great starting point for finding what you need.

Chapter 1: BGP Configuration

This chapter describes how to configure the Border Gateway Protocol on the Vyatta system.

The following topics are covered:

- BGP Overview
- Supported Standards
- Configuring BGP

BGP Overview

This section presents the following topics:

- iBGP and eBGP
- BGP Path Selection Process
- Scalability of BGP
- Route Flapping and Flap Damping
- AS Paths
- BGP Communities

Border Gateway Protocol (BGP) is the principal inter-domain routing protocol used on the Internet. BGP version 4 is specified in RFC 4271, which obsoletes the original BGPv4 specification defined in RFC 1771.

The principal concept of BGP is that of the Autonomous System (AS). An AS is a routing domain that is under one administrative authority, and which implements its own routing policies. For example, one Internet Service Provider (ISP) would have its own AS, while another would have its own, different, AS. Many large enterprises also have their own AS, particularly if they are multi-homed (that is, connected to multiple ISPs). The BGP routing protocol is used to convey network reachability information between ASs.

Routers that are configured to run BGP between one another are known as BGP peers or BGP neighbors. BGP uses a TCP connection on the well-known port 179 to exchange routing information between peers. BGP peers that are configured within the same AS are referred to as internal BGP (iBGP) peers. BGP peers that are configured in different ASs are referred to as external BGP (eBGP) peers.

There are two basic types of BGP route exchanges that occur between peers: route announcements and route withdrawals.

- A route announcement tells a peer that it can reach a particular network via the announcing router, and includes attributes associated with that path.
- A route withdrawal tells a peer that a previously announced route is no longer reachable via this peer.

All valid route announcements that are received on a BGP router are placed into the router’s BGP table. (These routes are typically referred to as BGP paths.) This means that, for a particular network prefix—for example, 10.0.0.0/8—the local BGP router might have recorded multiple available paths: one through any of its BGP peers. For each prefix, the BGP process uses a path selection algorithm to select the best available path from all those learned from its peers. Once the best path has been selected, that path becomes the candidate route from the BGP protocol for inserting into the active routing table.

Each BGP path includes several attributes that are used by the BGP path selection process to determine which path is the best available path. These attributes can also be used in user-defined routing policies applied to BGP; these can allow the router to perform additional actions on a matching path, such as determining whether to accept or reject a route announcement.

One of the most commonly used BGP path attributes is the AS path. The AS path lists each of the ASs by which the prefix has been announced, and is used to prevent routing loops. The AS path is read from right to left, where the right-most AS is the AS that originated the network prefix (that is, it was the first AS to announce reachability for this prefix). This AS is known as the origin AS.

As a network prefix is advertised between ASs, each AS prepends its own AS number to the AS path. For example, the AS path “4 3 2 1” indicates that AS 1 originated the network prefix. The prefix was advertised from AS 1 to AS 2, then from AS 2 to AS 3, and finally from AS 3 to AS 4.

Other BGP path attributes include origin, next hop, multi-exit discriminator (“med”), local preference (“local pref”), atomic aggregate, and aggregator. These attributes are described in more detail in another section of this document.

iBGP and eBGP

A BGP peer can be one of two types:

- Internal BGP (iBGP) peers are peers that are configured with the same AS number.
- External BGP (eBGP) peers are peers that are configured with different AS numbers.

iBGP

The BGP protocol requires that all iBGP peers within an AS have a connection to one another, creating a full-mesh of iBGP peering connections. (The exception to this is route reflection.) When a prefix is announced from one iBGP peer to another, the AS path is not changed. Due to the full-mesh requirement, all iBGP peers should have the same view of the BGP table, unless different routing policies have been applied to some of the peers.

When a router receives an iBGP announcement, the BGP process uses the BGP best path selection algorithm to determine whether the received announcement is the best available path for that prefix. If it is the best available path, then the BGP process uses this route as the BGP candidate route for insertion into the routing table, and the BGP process announces this path to all its peers, both iBGP and eBGP peers. If it is not the best available path, then the BGP process keeps a copy of this path in its BGP table, so that it can be used to calculate the best available path when path information for that prefix changes (for example, if the current best available path is withdrawn).

The BGP ID is a unique identifier in the format of an IP address used to identify a peer. The peering IP address is the actual IP address used for the BGP connection.

For iBGP peerings, the BGP ID and peering IP is frequently the IP address bound to that router's loopback interface. An iBGP session is usually contained within a local LAN, with multiple redundant physical links between the iBGP devices. For iBGP routes, reachability is all that is necessary, and the loopback interface is reachable so long as at least one physical interface is operational. Because of the physical and/or logical redundancy that exists between iBGP peers, iBGP peering on the loopback interface works well.

Since BGP does not provide reachability information, you must make sure that each iBGP peer knows how to reach other peers. To be able to reach one another, each peer must have some sort of Interior Gateway Protocol (IGP) route, such as a connected route, a static route, or a route through a dynamic routing protocol such as RIP or OSPF, which tells them how to reach the opposite router.

eBGP

External BGP is the method that different Autonomous Systems (ASs) use to interconnect with one another. eBGP usually takes place over WAN links, where there may be a single physical path between eBGP peers. Alternatively, they may have multiple eBGP peer connections to provide redundancy and/or traffic load balancing. Redundant peers use distinct BGP sessions so that, if one session fails, another can take over.

BGP uses an AS path to track the path of a prefix through the various ASs that send or receive the prefix announcement. When a prefix is announced to an eBGP peer, the local AS number is prepended to the AS path. This helps to prevent routing loops by rejecting any prefix announcements that include the local AS number in the AS path. Prefix announcements learned via eBGP are also analyzed using the BGP best path selection process.

For eBGP peerings, the BGP ID and peering IP address is typically the local IP address of the interface that is being used to connect to the eBGP peers. However if more than one physical interface is being used for eBGP peering it is also common to use a loopback IP address as the BGP ID, but still use the physical interface IP address as the peering IP address.

BGP Path Selection Process

The BGP process may receive advertisements from multiple peers for the same network prefix. Each of these announcements from a peer for a prefix is called a path. The BGP process selects the “best” path from all available paths and this path becomes the candidate route announced by BGP for inclusion in the Routing Information Base (RIB).

Depending on what other protocols also have candidate routes for this network prefix, the BGP route may or may not be added to the RIB. For instance if the RIB has candidate routes from both BGP and static routing for the same prefix, the static route, not the BGP route, will be included in the RIB. This is because the static route process has a lower administrative cost than the BGP process.

It is important to note that BGP will not consider paths that contain a NEXT_HOP value that is not reachable via an entry in the RIB. For all valid paths, the Vyatta system uses a BGP path selection process based on decision process described in RFC 4271, section 9.1. BGP paths are preferred based on the following:

- **LOCAL PREFERENCE**: Prefer the path with the highest LOCAL_PREF
- **AS PATH LENGTH**: Prefer the path with the shortest AS_PATH.
- **ORIGIN**: Prefer the path with the lowest ORIGIN type.
- **MULTI_EXIT_DISC**: Prefer the path with the lowest MED.
- **PEER TYPE**: Prefer paths learned via eBGP over paths learned via iBGP.
- **IGP METRIC**. Prefer paths with lower IGP metric for the path’s NEXT_HOP address.
- **BGP ID**: Prefer the path with the lowest BGP ID.
- **PEER IP**: Prefer the path with the lowest peer IP address.

The best path selection process is performed as “first match and out.” This means that two paths will be compared until the first difference in preference criteria. For example, two paths for the same network prefix may have the same LOCAL_PREF value, but different AS path lengths. In this case, the path with the shortest AS path would be the “best” path. If the peer IP address is being used to select the best path, this means that all other path criteria were the same for the available paths.

You can use the **show ip bgp** command to see the current best paths in the RIB.

Scalability of BGP

The Border Gateway Protocol 4 specification (RFC 4271) requires that iBGP peers be fully meshed; that is, every iBGP peer must have a connection to every other iBGP peer. A full mesh of iBGP peers does not scale well to large ASs, which can have hundreds of iBGP routers. To overcome scalability issues, two enhancements have been developed for BGP:

- BGP Confederations (RFC 3065)
- Route Reflection (RFC 2796)

Confederations

Confederations enable you to reduce the size and complexity of the iBGP mesh. In a BGP confederation, a single AS is divided into multiple internal sub-ASs to help keep the number of iBGP peer connections manageable. Each sub-AS is assigned its own AS number; this is typically assigned from the private AS number space, which ranges from 65412 to 65535. Within a sub-AS, all the standard iBGP rules, including full-mesh peering, apply. The connections between confederation sub-ASs use eBGP peering. One or more eBGP connections can be made between each sub-AS. The sub-ASs are grouped as a confederation, which advertises as a single AS to external peers.

Figure 1-1 shows the large number of iBGP connections that must be configured in even a moderately sized AS. In this example, 14 routers are participating in iBGP.

Figure 1-1 iBGP full mesh

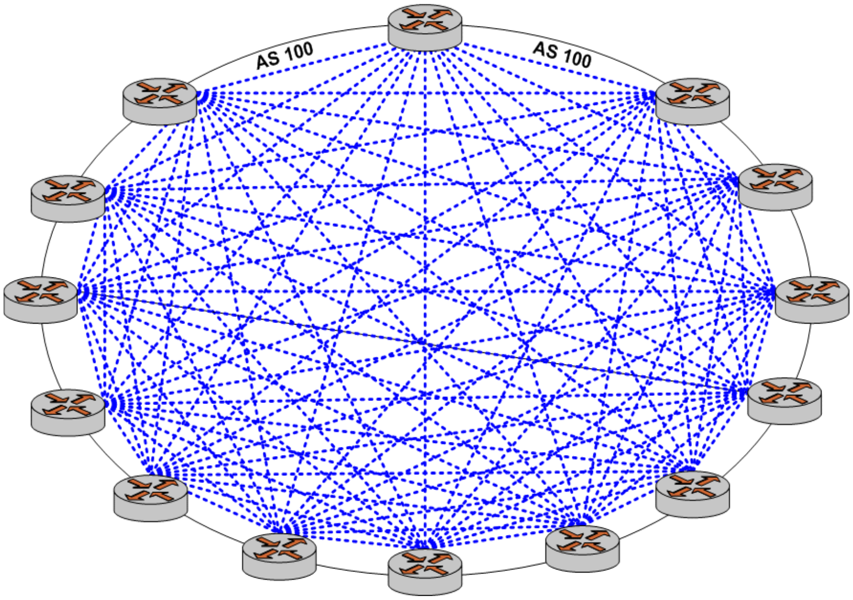
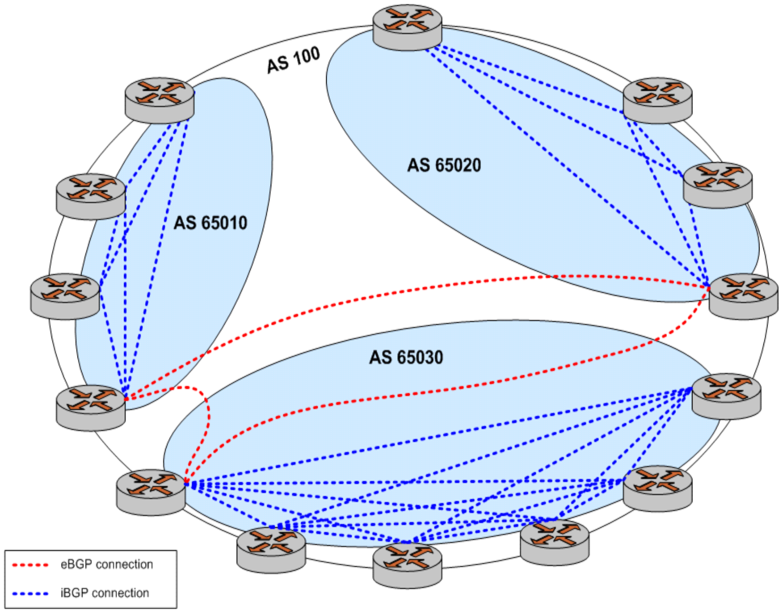


Figure 1-2 shows a BGP confederation that splits the single AS shown in Figure 1-1 into three sub-ASs, which each use private AS numbers. Within each sub-AS, all of the iBGP peers are fully meshed. The sub-ASs are connected to one another using an eBGP connection.

Figure 1-2 BGP confederation



Route Reflection

Another technology designed to help ASs with large numbers of iBGP peers is route reflection. In a standard BGP implementation, all iBGP peers must be fully meshed. because of this requirement, when an iBGP peer learns a route from another iBGP peer, the receiving router does not forward the route to any of its iBGP peers, since these routers should have learned the route directly from the announcing router.

In a route reflector environment the iBGP peers are no longer fully meshed. Instead, each iBGP peer has an iBGP connection to one or more route reflectors (RRs). Routers configured with a connection to an RR server are referred to as RR clients. Only the RR server is configured to be aware that the RR client is part of an RR configuration; from the RR client’s point of view, it is configured normally, and does not have any awareness that it is part of a RR configuration.

In route reflection, internal peers of an RR server are categorized into two types:

- **Client peers.** The RR server and its client peers form a cluster. Within a cluster, client peers need not be fully meshed, but must have an iBGP connection to at least one RR in the cluster.
- **Non-client peers.** Non-client peers, including the RR server, must be fully meshed.

An RR environment is unlike a regular environment, where iBGP peers never forward a route update to other iBGP peers (which is the reason why each iBGP peer must peer with all other peers). When an RR server receives an iBGP update from an RR client, these route updates can also be sent to all other RR clients. When an RR server receives a route update from a peer, it selects the best path based on its path selection rule. After the best path is selected, the RR server chooses its action depending on the type of the peer from which it learned the best path.

- If the route was learned from a client peer, the RR reflects the route to both client and non-client peers. All iBGP updates from client peers are reflected to all other client peers in the cluster. This is done regardless of whether the update was the best path for the RR itself.
- If the route was learned from a non-client iBGP peer, it is reflected out to all RR client peers.
- If the route was learned from an eBGP peer, the route is reflected to all RR clients and all non-clients.

Figure 1-3 shows again the full mesh of iBGP connections in even a moderately sized AS.

Figure 1-3 iBGP full mesh

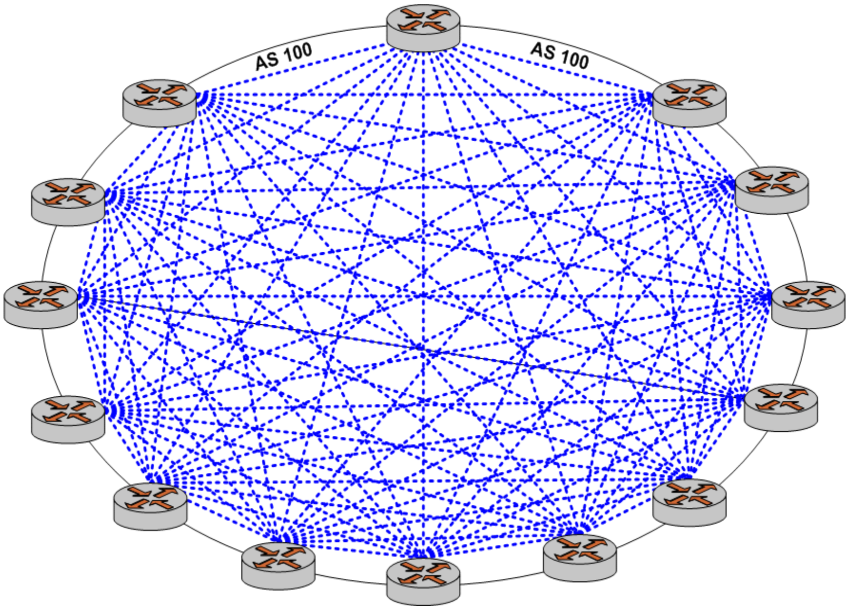
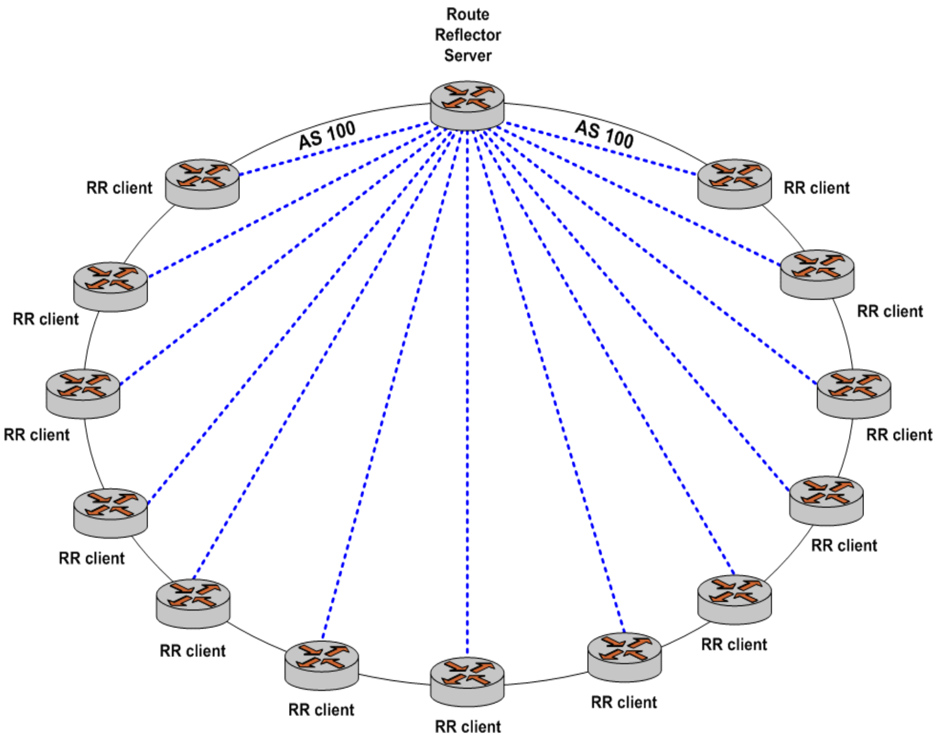


Figure 1-4 shows how introducing route reflection into the AS dramatically reduces the number of iBGP connections required within the AS.

Figure 1-4 iBGP route reflection



Note that to prevent looping, clients must not peer with RRs outside of the cluster.

To achieve redundancy, more than one RR server can be configured within a cluster. Also, to scale to very large networks, a large AS can be configured to have multiple clusters with redundant RR servers, where the RR servers are all configured with a full mesh of iBGP connections between the RR servers.

Route Flapping and Flap Damping

Route flapping is a situation where a route fluctuates repeatedly between being announced, then withdrawn, then announced, then withdrawn, and so on. In this situation, a BGP system will send an excessive number of update messages advertising network reachability information.

Route flapping can cause several different issues. First, each time a new route is learned or withdrawn the BGP best path selection process for that prefix must be executed, which can result in high CPU utilization. If a large enough number of routes are flapping, the BGP process may not be able to converge sufficiently quickly. Second, the route flapping issue can become amplified as it passes from peer to peer. For example, if a router with two peers flaps a route, and those two peers each have 10 peers, the flapping route affects 20 BGP routers.

Route dampening is intended to minimize the propagation of update messages between BGP peers for flapping routes. This reduces the load on these devices without unduly impacting the route convergence time for stable routes.

When route damping is enabled, a route is assigned a penalty each time it “flaps” (that is, each time it is announced and then withdrawn within a short interval). If the penalty exceeds 1000 (its *suppress* value) the route is suppressed.

After the route has been stable for a configured interval (its *half-life*) the penalty is reduced by half. Subsequently, the penalty is reduced every five seconds. When the penalty falls below a configured value (its *reuse* value), the route is unsuppressed.

The penalty applied to a route will never exceed the *maximum penalty*, which is computed from configured attributes as follows:

```
Maximum penalty = reuse * 2^(suppress/half-life)
```

While the route is being “damped,” updates and withdrawals for this route from a peer are ignored. This helps to localize the route flapping to a particular peering connection.

AS Paths

An AS path is a path to a destination in the Border Gateway Protocol (BGP). The path is represented as a sequence of AS numbers, which are the numbers uniquely identifying BGP autonomous systems. Each AS number represents an autonomous system (which may be comprised of multiple networks) that a packet traverses if it takes the associated route to the destination.

For a packet to reach a destination using this route, it traverses the listed ASs from the leftmost AS number to the rightmost, where the rightmost is the AS immediately preceding its destination.

Using policies, match conditions can be defined based on all or portions of the AS path. To do this, you can either specify the AS path directly in a policy command using a regular expression in the **as-path** attribute, or create a named AS path regular expression using the **as-path-list** attribute and including the name in a policy command.

BGP Communities

All BGP updates include a BGP attribute called the communities path attribute. The communities path attribute allows ASs to "tag" prefix announcements. This tag can then be used by routing policies to modify the normal behavior for that prefix announcement. For example, an AS could choose to filter out all prefix announcements containing a community value that identifies the prefix as a customer-generated prefix, instead only announcing the summary prefix for all customer prefixes. It is important to note that the community path attribute is carried in BGP update messages, which allows ASs not directly connected to each other to share information about a prefix.

The format for community identifiers is defined in RFC 1997: "BGP Communities Attribute." The community identifier is a 32-bit value, where the first two bytes of the value are the AS number and the second two bytes are an arbitrary value defined by the AS. This format can be represented as *AA:NN*, where *AA* is the AS number of the AS adding the community identifier to the community path attribute, and *NN* represents a user-defined policy value.

There are two types of BGP communities: "well-known" communities and user-defined or private communities.

The Vyatta router recognizes the following BGP well-known communities as per RFC 1997:

NO_EXPORT: All routes received carrying a communities attribute containing this value are not advertised outside a BGP confederation boundary (a stand-alone autonomous system that is not part of a confederation should be considered a confederation itself).

NO_ADVERTISE: All routes received carrying a communities attribute containing this value are not advertised to other BGP peers.

NO_EXPORT_SUBCONFED: All routes received carrying a communities attribute containing this value are not advertised to external BGP peers (this includes peers in other members autonomous systems inside a BGP confederation).

Supported Standards

The Vyatta implementation of BGP complies with the following standards:

- RFC 4271: BGP-4 Specification
- RFC 4273: Definitions of Managed Objects for BGP-4
- RFC 1997: BGP Communities Attribute
- RFC 3065: BGP Confederations RFC 3065
- RFC 2796: Route Reflection RFC 2796

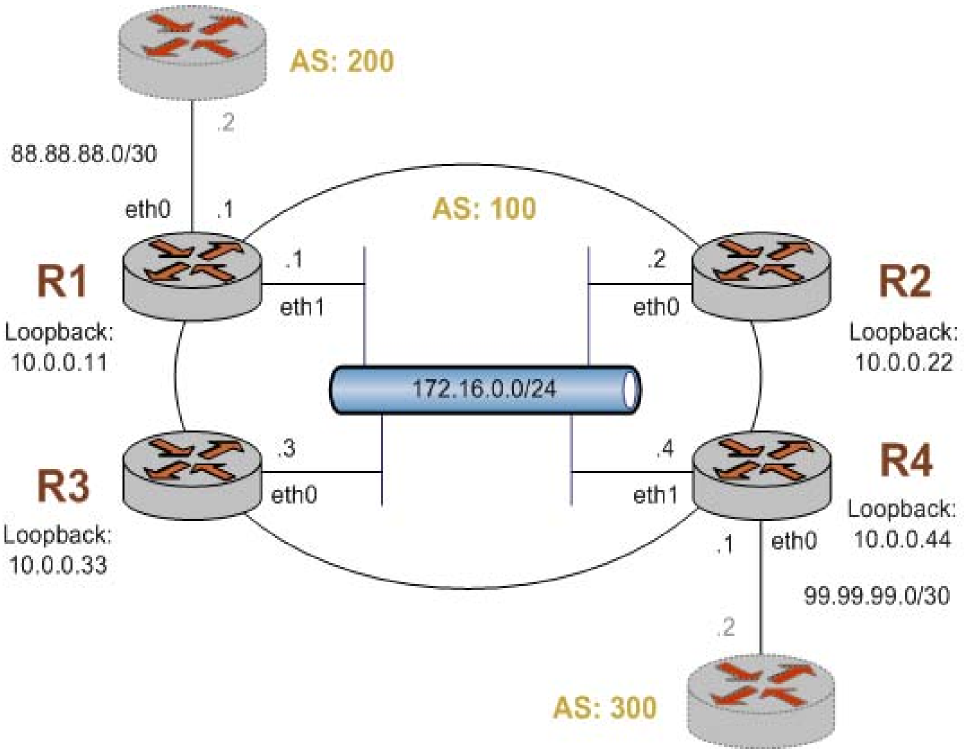
Configuring BGP

This section presents the following topics:

- Basic iBGP Configuration
- Verifying the iBGP Configuration
- Basic eBGP Configuration
- Verifying the eBGP Configuration
- Originating a Route to eBGP Neighbors
- Verifying the Route Origination
- Inbound Route Filtering
- Verifying the Inbound Filter
- Outbound Route Filtering
- Verifying the Outbound Filter
- Confederations
- Verifying the Confederation
- Route Reflectors
- Verifying the Route Reflector
- Route Redirection

This section presents sample configurations for BGP. The configuration examples are based on the reference diagram in Figure 1-5.

Figure 1-5 BGP configuration reference diagram



Basic iBGP Configuration

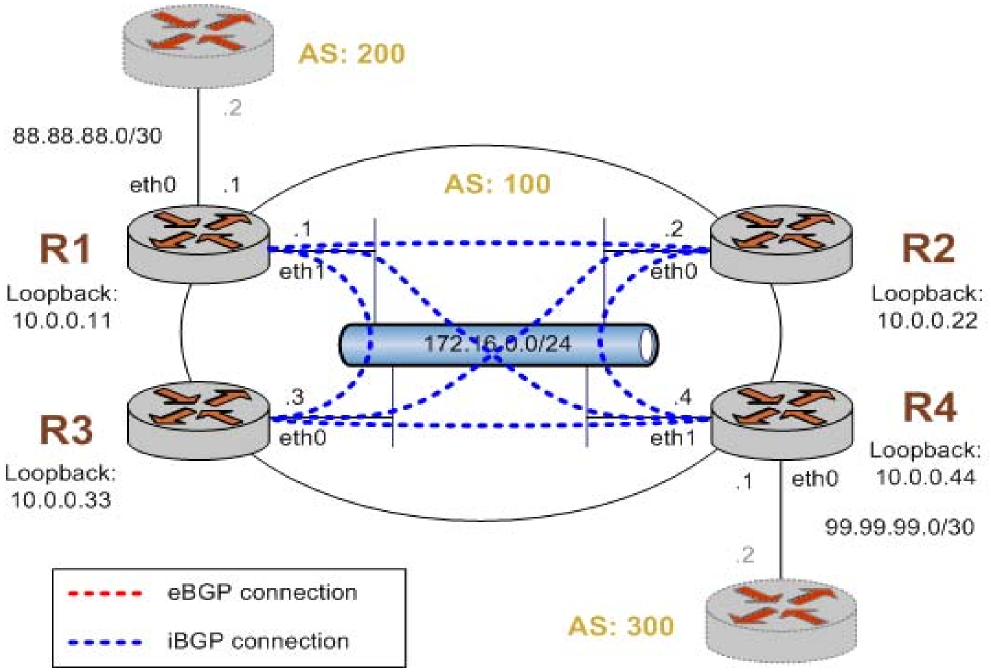
In this section, you configure iBGP on the routers labeled R1, R2, R3, and R4 in the reference network diagram. Each router has an iBGP peering connection to each of the other iBGP routers in the network, satisfying the full mesh iBGP peering requirement.

In the example the iBGP peering connections are established between iBGP neighbors using the loopback interface IP addresses. This is typical practice, particularly when there are redundant connections between the iBGP routers.

In order for the routers to be able to reach each other using the loopback IP address, the loopback IP addresses must be reachable via an entry in the router’s routing table. This requires some form of Internal Gateway Protocol (IGP). In the example here, we will use a basic Open Shortest Path First (OSPF) configuration to announce the loopback addresses between neighbors.

Figure 1-6 shows the BGP connections after you have completed the iBGP configuration.

Figure 1-6 Basic iBGP configuration



This example assumes that you have already configured the router interfaces; only the steps required to implement BGP are shown.

To create a basic iBGP configuration, perform the following steps in configuration mode:

Example 1-1 Basic iBGP configuration

Router	Step	Command(s)
R1	Advertize the loopback address within the OSPF area. This is needed for iBGP.	vyatta@R1# **set protocols ospf area 0.0.0.0 network 10.0.0.11/32** [edit]
R1	Advertize the local network within the OSPF area.	vyatta@R1# **set protocols ospf area 0.0.0.0 network 172.16.0.0/24** [edit]
R1	Advertize the external network within the OSPF area.	vyatta@R1# **set protocols ospf area 0.0.0.0 network 88.88.88.0/30** [edit]
R1	Set the router ID to be the loopback IP address.	vyatta@R1# **set protocols ospf parameters router-id 10.0.0.11** [edit]

Example 1-1 Basic iBGP configuration

R1	Configure the eth0 interface to be a passive interface (i.e. don’t advertize our internal network routes over the external network).	vyatta@R1# **set protocols ospf passive-interface eth0** [edit]
R1	Create an iBGP peer for R2. The peer is an iBGP peer because it resides within the same AS as this router.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.22 remote-as 100** [edit]
R1	Define the IP address on the local R1 router that is used to peer with the R2 router.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.22 update-source 10.0.0.11** [edit]
R1	Create an iBGP peer for R3. The peer is an iBGP peer because it resides within the same AS as this router.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.33 remote-as 100** [edit]
R1	Define the IP address on the local R1 router that is used to peer with the R3 router.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.33 update-source 10.0.0.11** [edit]
R1	Create an iBGP peer for R4. The peer is an iBGP peer because it resides within the same AS as this router.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.44 remote-as 100** [edit]
R1	Define the IP address on the local R1 router that is used to peer with the R4 router.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.44 update-source 10.0.0.11** [edit]
R1	Set the router ID to the loopback address, which on R1 is 10.0.0.11.	vyatta@R1# **set protocols bgp 100 parameters router-id 10.0.0.11** [edit]
R1	Commit the configuration.	vyatta@R1# **commit** [edit]

Example 1-1 Basic iBGP configuration

<table>
<tr>
<td>R1</td>
<td>Display the configuration.</td>
<td>

```
vyatta@R1# show protocols
 bgp 100 {
     neighbor 10.0.0.22 {
         remote-as 100
         update-source 10.0.0.11
     }
     neighbor 10.0.0.33 {
         remote-as 100
         update-source 10.0.0.11
     }
     neighbor 10.0.0.44 {
         remote-as 100
         update-source 10.0.0.11
     }
     parameters {
         router-id 10.0.0.11
     }
 }
 ospf {
     area 0.0.0.0 {
         network 172.16.0.0/24
         network 88.88.88.0/30
         network 10.0.0.11/32
     }
     parameters {
         router-id 10.0.0.11
     }
     passive-interface eth0
 }
[edit]
```

</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>Advertize the loopback address within the OSPF area. This is needed for iBGP.</td>
<td>

```
vyatta@R2# set protocols ospf area 0.0.0.0 network
10.0.0.22/32
[edit]
```

</td>
</tr>
<tr>
<td>R2</td>
<td>Advertize the local network within the OSPF area.</td>
<td>

```
vyatta@R2# set protocols ospf area 0.0.0.0 network
172.16.0.0/24
[edit]
```

</td>
</tr>
<tr>
<td>R2</td>
<td>Set the router ID to be the loopback IP address.</td>
<td>

```
vyatta@R2# set protocols ospf parameters router-id
10.0.0.22
[edit]
```

</td>
</tr>
</table>

Example 1-1 Basic iBGP configuration

R2	Create an iBGP peer for R1. The peer is an iBGP peer because it resides within the same AS as this router.	vyatta@R2# **set protocols bgp 100 neighbor 10.0.0.11 remote-as 100** [edit]
R2	Define the IP address on the local R2 router that is used to peer with the R1 router.	vyatta@R2# **set protocols bgp 100 neighbor 10.0.0.11 update-source 10.0.0.22** [edit]
R2	Create an iBGP peer for R3. The peer is an iBGP peer because it resides within the same AS as this router.	vyatta@R2# **set protocols bgp 100 neighbor 10.0.0.33 remote-as 100** [edit]
R2	Define the IP address on the local R2 router that is used to peer with the R3 router.	vyatta@R2# **set protocols bgp 100 neighbor 10.0.0.33 update-source 10.0.0.22** [edit]
R2	Create an iBGP peer for R4. The peer is an iBGP peer because it resides within the same AS as this router.	vyatta@R2# **set protocols bgp 100 neighbor 10.0.0.44 remote-as 100** [edit]
R2	Define the IP address on the local R2 router that is used to peer with the R4 router.	vyatta@R2# **set protocols bgp 100 neighbor 10.0.0.44 update-source 10.0.0.22** [edit]
R2	Set the router ID to the loopback address, which on R2 is 10.0.0.22.	vyatta@R2# **set protocols bgp 100 parameters router-id 10.0.0.22** [edit]

Example 1-1 Basic iBGP configuration

<table>
<tr><td>R2</td><td>Display the configuration.</td><td><pre>vyatta@R2# <b>show protocols</b>
 bgp 100 {
     neighbor 10.0.0.11 {
         remote-as 100
         update-source 10.0.0.22
     }
     neighbor 10.0.0.33 {
         remote-as 100
         update-source 10.0.0.22
     }
     neighbor 10.0.0.44 {
         remote-as 100
         update-source 10.0.0.22
     }
     parameters {
         router-id 10.0.0.22
     }
 }
 ospf {
     area 0.0.0.0 {
         network 172.16.0.0/24
         network 10.0.0.22/32
     }
     parameters {
         router-id 10.0.0.22
     }
 }
[edit]</pre></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>R3</td><td>Advertize the loopback address within the OSPF area. This is needed for iBGP.</td><td><pre>vyatta@R3# <b>set protocols ospf area 0.0.0.0 network
10.0.0.33/32</b>
[edit]</pre></td></tr>
<tr><td>R3</td><td>Advertize the local network within the OSPF area.</td><td><pre>vyatta@R3# <b>set protocols ospf area 0.0.0.0 network
172.16.0.0/24</b>
[edit]</pre></td></tr>
<tr><td>R3</td><td>Set the router ID to be the loopback IP address.</td><td><pre>vyatta@R3# <b>set protocols ospf parameters router-id
10.0.0.33</b>
[edit]</pre></td></tr>
<tr><td>R3</td><td>Create an iBGP peer for R1. The peer is an iBGP peer because it resides within the same AS as this router.</td><td><pre>vyatta@R3# <b>set protocols bgp 100 neighbor 10.0.0.11
remote-as 100</b>
[edit]</pre></td></tr>
</table>

Example 1-1 Basic iBGP configuration

R3	Define the IP address on the local R3 router that is used to peer with the R1 router.	vyatta@R3# **set protocols bgp 100 neighbor 10.0.0.11 update-source 10.0.0.33** [edit]
R3	Create an iBGP peer for R2. The peer is an iBGP peer because it resides within the same AS as this router.	vyatta@R3# **set protocols bgp 100 neighbor 10.0.0.22 remote-as 100** [edit]
R3	Define the IP address on the local R3 router that is used to peer with the R2 router.	vyatta@R3# **set protocols bgp 100 neighbor 10.0.0.22 update-source 10.0.0.33** [edit]
R3	Create an iBGP peer for R4. The peer is an iBGP peer because it resides within the same AS as this router.	vyatta@R3# **set protocols bgp 100 neighbor 10.0.0.44 remote-as 100** [edit]
R3	Define the IP address on the local R3 router that is used to peer with the R4 router.	vyatta@R3# **set protocols bgp 100 neighbor 10.0.0.44 update-source 10.0.0.33** [edit]
R3	Set the router ID to the loopback address, which on R3 is 10.0.0.33.	vyatta@R3# **set protocols bgp 100 parameters router-id 10.0.0.33** [edit]
R3	Commit the configuration.	vyatta@R3# **commit** [edit]

Example 1-1 Basic iBGP configuration

<table>
<tr><td>R3</td><td>Display the configuration.</td><td><pre>
vyatta@R3# show protocols
 bgp 100 {
     neighbor 10.0.0.11 {
         remote-as 100
         update-source 10.0.0.33
     }
     neighbor 10.0.0.22 {
         remote-as 100
         update-source 10.0.0.33
     }
     neighbor 10.0.0.44 {
         remote-as 100
         update-source 10.0.0.33
     }
     parameters {
         router-id 10.0.0.33
     }
 }
 ospf {
     area 0.0.0.0 {
         network 172.16.0.0/24
         network 10.0.0.33/32
     }
     parameters {
         router-id 10.0.0.33
     }
 }
[edit]
</pre></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>R4</td><td>Advertize the loopback address within the OSPF area. This is needed for iBGP.</td><td><pre>
vyatta@R4# set protocols ospf area 0.0.0.0 network
10.0.0.44/32
[edit]
</pre></td></tr>
<tr><td>R4</td><td>Advertize the local network within the OSPF area.</td><td><pre>
vyatta@R4# set protocols ospf area 0.0.0.0 network
172.16.0.0/24
[edit]
</pre></td></tr>
<tr><td>R4</td><td>Advertize the external network within the OSPF area.</td><td><pre>
vyatta@R4# set protocols ospf area 0.0.0.0 network
99.99.99.0/30
[edit]
</pre></td></tr>
<tr><td>R4</td><td>Set the router ID to be the loopback IP address.</td><td><pre>
vyatta@R4# set protocols ospf parameters router-id
10.0.0.44
[edit]
</pre></td></tr>
</table>

Example 1-1 Basic iBGP configuration

R4	Configure the eth0 interface to be a passive interface (i.e. don’t advertize our internal network routes over the external network).	vyatta@R4# **set protocols ospf passive-interface eth0** [edit]
R4	Create an iBGP peer for R1. The peer is an iBGP peer because it resides within the same AS as this router.	vyatta@R4# **set protocols bgp 100 neighbor 10.0.0.11 remote-as 100** [edit]
R4	Define the IP address on the local R4 router that is used to peer with the R1 router.	vyatta@R4# **set protocols bgp 100 neighbor 10.0.0.11 update-source 10.0.0.44** [edit]
R4	Create an iBGP peer for R2. The peer is an iBGP peer because it resides within the same AS as this router.	vyatta@R4# **set protocols bgp 100 neighbor 10.0.0.22 remote-as 100** [edit]
R4	Define the IP address on the local R4 router that is used to peer with the R2 router.	vyatta@R4# **set protocols bgp 100 neighbor 10.0.0.22 update-source 10.0.0.44** [edit]
R4	Create an iBGP peer for R3. The peer is an iBGP peer because it resides within the same AS as this router.	vyatta@R4# **set protocols bgp 100 neighbor 10.0.0.33 remote-as 100** [edit]
R4	Define the IP address on the local R4 router that is used to peer with the R3 router.	vyatta@R4# **set protocols bgp 100 neighbor 10.0.0.33 update-source 10.0.0.44** [edit]
R4	Set the router ID to the loopback address, which on R4 is 10.0.0.44.	vyatta@R4# **set protocols bgp 100 parameters router-id 10.0.0.44** [edit]
R4	Commit the configuration.	vyatta@R4# **commit** [edit]

Example 1-1 Basic iBGP configuration

R4 Display the configuration.

```
vyatta@R4# show protocols
 bgp 100 {
     neighbor 10.0.0.11 {
         remote-as 100
         update-source 10.0.0.44
     }
     neighbor 10.0.0.22 {
         remote-as 100
         update-source 10.0.0.44
     }
     neighbor 10.0.0.33 {
         remote-as 100
         update-source 10.0.0.44
     }
     parameters {
         router-id 10.0.0.44
     }
 }
 ospf {
     area 0.0.0.0 {
         network 172.16.0.0/24
         network 99.99.99.0/30
         network 10.0.0.44/32
     }
     parameters {
         router-id 10.0.0.44
     }
     passive-interface eth0
 }
[edit]
```

Verifying the iBGP Configuration

The following commands can be used to verify the iBGP configuration.

R1: show ip bgp summary

Example 1-2 shows the output of the **show ip bgp summary** command for router R1 at this stage of the configuration.

Example 1-2 Verifying iBGP on R1: ”show ip bgp summary”

```
vyatta@R1>vyatta@R1:~$ show ip bgp summary
```

```
BGP router identifier 10.0.0.11, local AS number 100
RIB entries 1, using 64 bytes of memory
Peers 3, using 7560 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.22       4   100       6      10        0    0    0 00:04:18        0
10.0.0.33       4   100       6      10        0    0    0 00:04:14        0
10.0.0.44       4   100       5       6        0    0    0 00:02:55        0

Total number of neighbors 3
vyatta@R1:~$
```

The most important fields in the output for **show ip bgp summary** are the **Up/Down** and **State** fields. All the iBGP peers for R1 show times in the **Up/Down** field, which means the connection has been “established” for that period of time. The established state indicates that the peers have successfully created a BGP connection between one another, and are now able to send and receive BGP update messages.

If a peer shows in either **Active** or **Idle** in the **State** field, it means there is some issue that is keeping the BGP peers from forming an adjacency.

- The **Active** state identifies that the local router is actively trying to establish a TCP connection to the remote peer. You may see this if the local peer has been configured, but the remote peer is unreachable or has not been configured.
- The **Idle** state indicates that the local router has not allocated any resources for that peer connection, so any incoming connection requests will be refused.

R1: show ip bgp

Because we have not configured any routing announcements yet, the BGP table is currently empty. This can be seen by the output of **show ip bgp** for R1, which is shown in Example 1-3.

Example 1-3 Verifying iBGP on R1: “show ip bgp”

```
vyatta@R1:~$ show ip bgp
No BGP network exists
vyatta@R1:~$
```

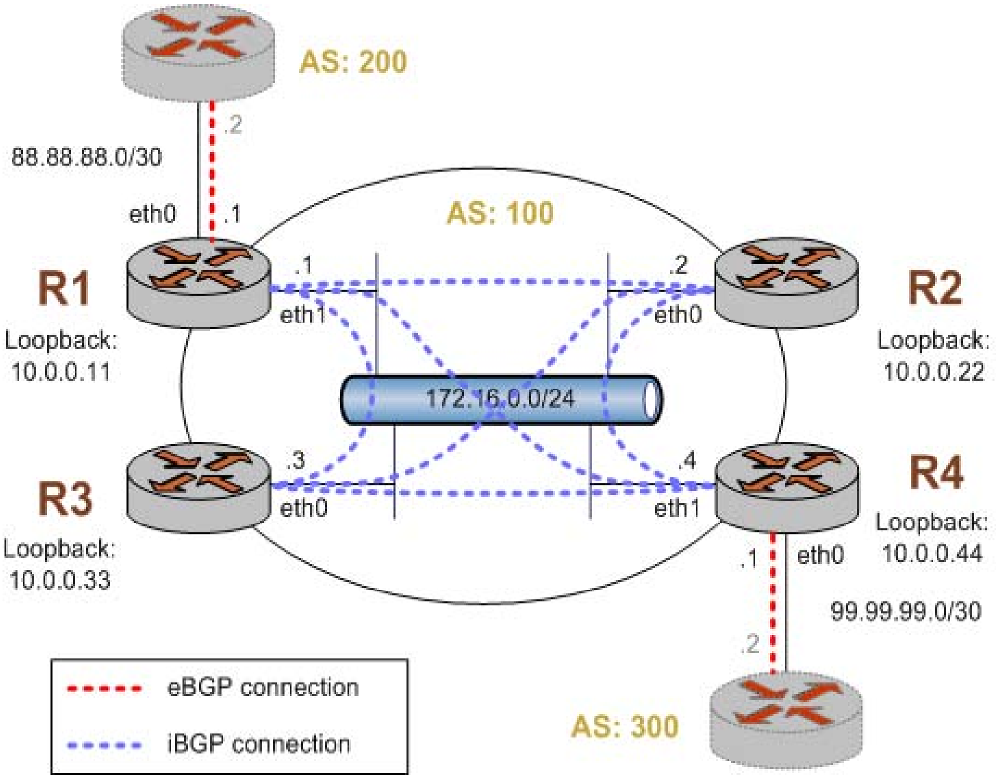
Basic eBGP Configuration

In this section, you configure eBGP on the routers labeled R1 and R4 in the reference network diagram. Router R1 is peering with an eBGP neighbor that is configured to be in AS 200 and router R4 is peering with an eBGP neighbor in AS 300.

In this example, the eBGP peering connections are established between eBGP neighbors using the physical interface IP addresses. This is a common configuration for eBGP peers. If the link between the peers goes down, the peering relationship should also go down since there is no redundancy.

After the basic eBGP configuration has been completed, the network should look as shown in Figure 1-7.

Figure 1-7 Basic eBGP configuration



This example assumes the following:

- The configuration in Example 1-6 has already been performed.
- The eBGP peers connecting to R1 and R4 have been properly configured for BGP.

To create a basic eBGP configuration, perform the following steps in configuration mode:

Example 1-4 Basic eBGP configuration

Router	Step	Command(s)
R1	Create an eBGP peer for R1. The peer is an eBGP peer because it resides in a different AS than this router.	vyatta@R1# **set protocols bgp neighbor 88.88.88.2 remote-as 200** [edit]
R1	Commit the configuration.	vyatta@R1# **commit** [edit]
R4	Create an eBGP peer for R4. The peer is an eBGP peer because it resides in a different AS than this router.	vyatta@R4# **set protocols bgp neighbor 99.99.99.2 remote-as 300** [edit]
R4	Commit the configuration.	vyatta@R4# **commit** [edit]

Verifying the eBGP Configuration

The following commands can be used to verify the eBGP configuration. Note that the output shown for these commands would be obtained *after* the configuration for both router R1 and router R4 has been completed.

R1: show ip bgp summary

Example 1-5 shows the output of the **show ip bgp summary** command for router R1 at this stage of the configuration.

Example 1-5 Verifying eBGP on R1: “show ip bgp summary”

```
vyatta@R1:~$ show ip bgp summary
BGP router identifier 10.0.0.11, local AS number 100
RIB entries 23, using 1472 bytes of memory
Peers 4, using 10080 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.22       4   100      40      44        0    0    0 00:38:23         0
10.0.0.33       4   100      40      44        0    0    0 00:38:22         0
```

```
10.0.0.44       4   100      43      47        0    0    0 00:06:51        7
88.88.88.2      4   200       4       5        0    0    0 00:01:22        5

Total number of neighbors 4
vyatta@R1:~$
```

After adding the eBGP peer 88.88.88.2 (the BGP ID configured for the router connected to AS 200) to R1 we can see that the connection to the new peer is established (in the **Up/Down** field). This indicates that the peer was properly preconfigured for this connection.

Additionally you may notice that the **MsgRcvd** and **MsgSent** fields for peer 88.88.88.2 shows “4” and “5” respectively. This shows that R1 has received four BGP messages from 88.88.88.2 and sent it five, which are associated with the connection to the peer (the **show ip bgp neighbors** command can be used to provide additional detail). Also, you can see that router R1 has received seven route prefixes (**PfxRcd** column) from 10.0.0.44 and five route prefixes from 88.88.88.2. The prefixes received can be seen using the **show ip bgp** command as follows.

R1: show ip bgp

Example 1-6 shows the output of the **show ip bgp** command for router R1 at this stage of the configuration.

Example 1-6 Verifying eBGP on R1: “show ip bgp”

```
vyatta@R1:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 2.0.0.0/24       88.88.88.2               0             0 200 i
*> 2.1.0.0/24       88.88.88.2               0             0 200 i
*> 2.2.0.0/24       88.88.88.2               0             0 200 i
*>i3.0.0.0/24       99.99.99.2               0    100      0 300 i
*>i3.1.0.0/24       99.99.99.2               0    100      0 300 i
*>i3.2.0.0/24       99.99.99.2               0    100      0 300 i
*> 12.0.0.0         88.88.88.2               0             0 200 i
*>i13.0.0.0/24      99.99.99.2               0    100      0 300 i
*> 88.88.88.0/30    88.88.88.2               0             0 200 i
*>i99.99.99.0/30    99.99.99.2               0    100      0 300 i
*>i172.16.128.0/24  99.99.99.2               0    100      0 300 i
*>i192.168.2.0      99.99.99.2               0    100      0 300 i
```

```
Total number of prefixes 12
vyatta@R1:~$
```

From this output we can see that R1 knows about twelve prefixes within BGP, five from AS200 and seven from AS300.

There are two symbols that are displayed at the beginning of each of the prefixes shown in the output of **show ip bgp**. The first symbol is the status code for a valid route, which is an asterisk ("*"). Essentially all routes shown in the BGP table should be preceded by this symbol. The second symbol is the greater than character (">"), which indicates which path is the best available path as determined by the BGP best path selection process. The **show ip bgp** command shows only the best path to each peer.

R1: show ip route bgp

Example 1-7 shows the output of the **show ip route bgp** command for router R1 at this stage of the configuration.

Example 1-7 Verifying eBGP on R1: "show ip route bgp"

```
vyatta@R1:~$ show ip route bgp
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

B>* 2.0.0.0/24 [20/0] via 88.88.88.2, eth0, 00:06:28
B>* 2.1.0.0/24 [20/0] via 88.88.88.2, eth0, 00:06:28
B>* 2.2.0.0/24 [20/0] via 88.88.88.2, eth0, 00:06:28
B>* 3.0.0.0/24 [200/0] via 99.99.99.2, eth1 (recursive via 172.16.0.4), 00:06:56
B>* 3.1.0.0/24 [200/0] via 99.99.99.2, eth1 (recursive via 172.16.0.4), 00:06:56
B>* 3.2.0.0/24 [200/0] via 99.99.99.2, eth1 (recursive via 172.16.0.4), 00:06:56
B>* 12.0.0.0/8 [20/0] via 88.88.88.2, eth0, 00:06:28
B>* 13.0.0.0/24 [200/0] via 99.99.99.2, eth1 (recursive via 172.16.0.4), 00:06:56
B   88.88.88.0/30 [20/0] via 88.88.88.2 inactive, 00:06:28
B   99.99.99.0/30 [200/0] via 99.99.99.2 inactive, 00:06:56
B>* 172.16.128.0/24 [200/0] via 99.99.99.2, eth1 (recursive via 172.16.0.4),
00:06:56
B>* 192.168.2.0/24 [200/0] via 99.99.99.2, eth1 (recursive via 172.16.0.4),
00:06:56
vyatta@R1:~$
```

The **show ip route bgp** command displays the routes in the RIB that were learned via BGP. This is different from the output of **show ip bgp**, which shows all paths learned via BGP regardless of whether it is the best BGP path and whether the BGP candidate to the RIB for the prefix is the best route (for example, it has the lowest admin cost).

The output for the same operational BGP commands run on router R4 yields similar results.

R4: show ip bgp summary

Example 1-8 shows the output of the **show ip bgp summary** command for router R4 at this stage of the configuration.

Example 1-8 Verifying eBGP on R4: “show ip bgp summary”

```
vyatta@R4:~$ show ip bgp summary
BGP router identifier 10.0.0.44, local AS number 100
RIB entries 23, using 1472 bytes of memory
Peers 4, using 10080 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.11       4   100     511     512        0    0    0 00:13:01         5
10.0.0.22       4   100     495     507        0    0    0 08:12:22         0
10.0.0.33       4   100     492     511        0    0    0 08:01:00         0
99.99.99.2      4   300      11      12        0    0    0 00:08:03         7

Total number of neighbors 4
vyatta@R4:~$
```

R4: show ip bgp

Example 1-9 shows the output of the **show ip bgp** command for router R4 at this stage of the configuration.

Example 1-9 Verifying eBGP on R4: “show ip bgp ”

```
vyatta@R4:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.44
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i2.0.0.0/24       88.88.88.2               0    100      0 200 i
*>i2.1.0.0/24       88.88.88.2               0    100      0 200 i
*>i2.2.0.0/24       88.88.88.2               0    100      0 200 i
*> 3.0.0.0/24       99.99.99.2               0             0 300 i
*> 3.1.0.0/24       99.99.99.2               0             0 300 i
*> 3.2.0.0/24       99.99.99.2               0             0 300 i
*>i12.0.0.0         88.88.88.2               0    100      0 200 i
```

```
*> 13.0.0.0/24      99.99.99.2               0             0 300 i
*>i88.88.88.0/30    88.88.88.2               0     100     0 200 i
*> 99.99.99.0/30    99.99.99.2               0             0 300 i
*> 172.16.128.0/24  99.99.99.2               0             0 300 i
*> 192.168.2.0      99.99.99.2               0             0 300 i

Total number of prefixes 12
vyatta@R4:~$
```

Router R4's BGP table contains the paths it learned from its eBGP peer, as well as the paths it learned from its iBGP neighbor R1.

Originating a Route to eBGP Neighbors

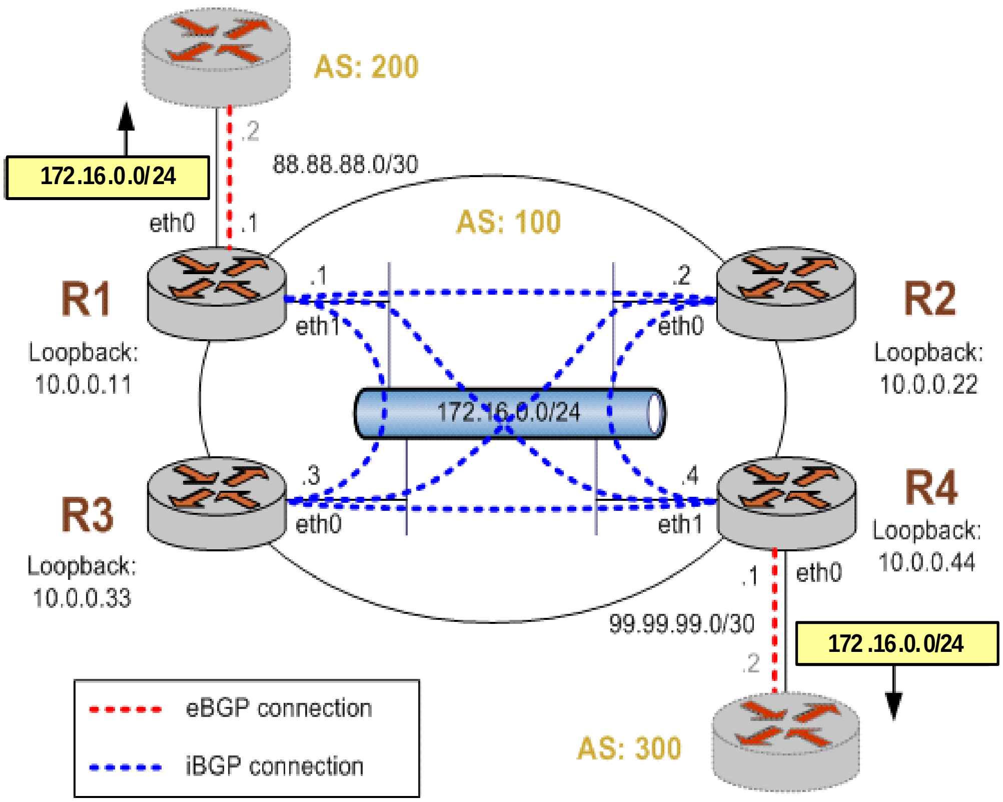
One of the common requirements for BGP configurations is to originate a network prefix to BGP peers. On the Vyatta router this is accomplished using the **network** option within the BGP configuration.

In this section, you originate the network prefix from both the R1 and R4 routers. This is shown in Figure 1-8.

NOTE *We assume that the routers in AS200 and AS300 are configured appropriately as eBGP peers.*

NOTE *The example in this section assumes that the desired network to originate to our BGP peers is 172.16.0.0/24, which is a private RFC 1918 network address. Typically, the router would be originating a public IP network assigned by an Internet routing registry such as the American Registry for Internet Numbers (ARIN).*

Figure 1-8 Originating a route to eBGP neighbors



This example assumes that the configurations in previous sections have been performed.

To originate a route to eBGP neighbors, perform the following steps in configuration mode:

Example 1-10 Originating routes to eBGP neighbors

Router	Step	Command(s)
R1	Advertize the local network to BGP.	vyatta@R1# **set protocols bgp 100 network 172.16.0.0/24** [edit]
R1	Commit the configuration.	vyatta@R1# **commit** [edit]
R4	Advertize the local network to BGP.	vyatta@R4# **set protocols bgp 100 network 172.16.0.0/24** [edit]
R4	Commit the configuration.	vyatta@R4# **commit** [edit]

Verifying the Route Origination

The following commands can be used to verify the route origination configuration.

R1: show ip bgp summary

Example 1-11 shows the output of the **show ip bgp summary** command for router R1 at this stage of the configuration. The **MsgSent** column indicates that the router has been sending BGP messages, showing the number of BGP messages that have been sent to each peer.

Example 1-11 Verifying route origination on R1: “show ip bgp summary”

```
vyatta@R1:~$ show ip bgp summary
BGP router identifier 10.0.0.11, local AS number 100
RIB entries 25, using 1600 bytes of memory
Peers 4, using 10080 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.22       4   100      50      55        0    0    0 00:48:02        0
10.0.0.33       4   100      50      55        0    0    0 00:48:01        0
10.0.0.44       4   100      54      58        0    0    0 00:16:30        8
88.88.88.2      4   200      14      17        0    0    0 00:11:01        5

Total number of neighbors 4
vyatta@R1:~$
```

R1: show ip bgp

Example 1-12 shows the output of the **show ip bgp** command for router R1 at this stage of the configuration.The AS column shows the AS path. You can check this column to make sure you received the route that you expected from each BGP neighbor.

Example 1-12 Verifying route origination on R1: “show bgp neighbor-routes”

```
vyatta@R1:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 2.0.0.0/24       88.88.88.2               0             0 200 i
*> 2.1.0.0/24       88.88.88.2               0             0 200 i
*> 2.2.0.0/24       88.88.88.2               0             0 200 i
*>i3.0.0.0/24       99.99.99.2               0    100      0 300 i
*>i3.1.0.0/24       99.99.99.2               0    100      0 300 i
*>i3.2.0.0/24       99.99.99.2               0    100      0 300 i
*> 12.0.0.0         88.88.88.2               0             0 200 i
*>i13.0.0.0/24      99.99.99.2               0    100      0 300 i
*> 88.88.88.0/30    88.88.88.2               0             0 200 i
*>i99.99.99.0/30    99.99.99.2               0    100      0 300 i
*> 172.16.0.0/24    0.0.0.0                  1         32768 i
* i                 10.0.0.44                1    100      0 i
*>i172.16.128.0/24  99.99.99.2               0    100      0 300 i
*>i192.168.2.0      99.99.99.2               0    100      0 300 i

Total number of prefixes 13
vyatta@R1:~$
```

Router R1 shows the prefix 172.16.0.0/24 from peer 0.0.0.0 which is used to represent itself and it also shows the 172.16.0.0/24 prefix being learned from iBGP neighbor R4. Note that the AS Path for locally originated routes does not contain any AS numbers in the **Path** column, this is because the originating AS (AS 100) will be added to BGP updates sent to eBGP peers.

This is also the first case where we have more than one path available for a prefix (172.16.0.0/24) since it is being originated on both R1 and R4. In this case, on R1, the route originated on R1 is the best path and has been selected as such by the BGP best path selection process. We can see that there is a second path available via R4. If for some reason the locally originated route is withdrawn from the BGP table, then the path via R4 would become the best path available.

AS 200: show ip bgp

Example 1-13 shows the output of the **show ip bgp** command for the router in AS 200 at this stage of the configuration.

Example 1-13 Verifying route origination in AS 200: “show ip bgp”

```
vyatta@AS200:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 2.0.0.0/24       0.0.0.0                  0         32768 i
*> 2.1.0.0/24       0.0.0.0                  0         32768 i
*> 2.2.0.0/24       0.0.0.0                  0         32768 i
*> 3.0.0.0/24       88.88.88.1                             0 100 300 i
*> 3.1.0.0/24       88.88.88.1                             0 100 300 i
*> 3.2.0.0/24       88.88.88.1                             0 100 300 i
*> 12.0.0.0         0.0.0.0                  0         32768 i
*> 13.0.0.0/24      88.88.88.1                             0 100 300 i
*> 88.88.88.0/30    0.0.0.0                  0         32768 i
*> 99.99.99.0/30    88.88.88.1                             0 100 300 i
*> 172.16.0.0/24    88.88.88.1               1             0 100 i
*> 172.16.128.0/24  88.88.88.1                             0 100 300 i
*> 192.168.2.0      88.88.88.1                             0 100 300 i

Total number of prefixes 13
vyatta@AS200:~$
```

We can confirm that router R1’s peer is learning the route via BGP by checking the BGP table on that router. Note that normally you would not be able to do this, since the router is in a different AS and therefore is probably controlled by a different organization. The 172.16.0.0/24 prefix is accessed via 88.88.88.1 (R1), and, because its path starts there, has an origin AS of 100.

Inbound Route Filtering

Another common requirement for BGP configurations is to filter inbound routing announcements from a BGP peer. On the Vyatta system this is accomplished using routing policies that are then applied to the BGP process as “import” policies.

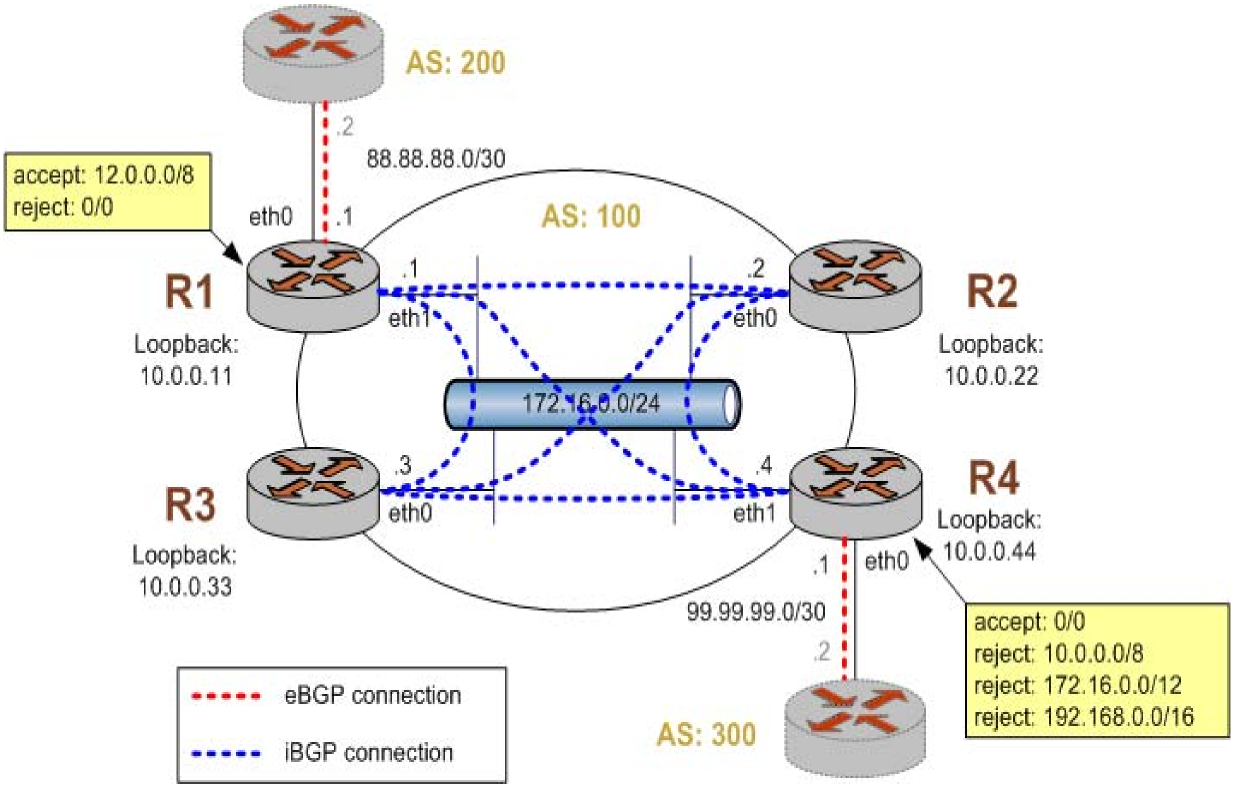
Example 1-14 creates the following inbound filtering policies:

- R1 should only accept network 12.0.0.0/8 from its eBGP peer, and reject everything else.
- R4 should allow all Internet routes, but reject all RFC 1918 networks from its eBGP peer.

This import policy is shown in Figure 1-9.

NOTE *We assume that the routers in AS200 and AS300 are configured appropriately as eBGP peers.*

Figure 1-9 Filtering inbound routes



To create this inbound route filter, perform the following steps in configuration mode:

Example 1-14 Creating an import policy

Router	Step	Command(s)
R1	Create a list of prefixes to allow. In this case we just have one - 12.0.0.0/8.	vyatta@R1# **set policy prefix-list ALLOW-PREFIXES rule 1 action permit** [edit] vyatta@R1# **set policy prefix-list ALLOW-PREFIXES rule 1 prefix 12.0.0.0/8** [edit]
R1	Create a route map rule to permit all prefixes in our list.	vyatta@R1# **set policy route-map eBGP-IMPORT rule 10 action permit** [edit] vyatta@R1# **set policy route-map eBGP-IMPORT rule 10 match ip address prefix-list ALLOW-PREFIXES** [edit]
R1	Create a route map rule to deny all other prefixes.	vyatta@R1# **set policy route-map eBGP-IMPORT rule 20 action deny** [edit]

Example 1-14 Creating an import policy

<table>
<tr><td>R1</td><td>Assign the route map policy created as the import route map policy for AS 200.</td><td><pre>vyatta@R1# <b>set protocols bgp 100 neighbor 88.88.88.2
route-map import eBGP-IMPORT</b>
[edit]</pre></td></tr>
<tr><td>R1</td><td>Commit the configuration.</td><td><pre>vyatta@R1# <b>commit</b>
[edit]</pre></td></tr>
<tr><td>R1</td><td>Reset the BGP session to the peer so that the new policies are enabled.</td><td><pre>vyatta@R1# <b>run clear ip bgp 88.88.88.2</b>
[edit]</pre></td></tr>
<tr><td>R1</td><td>Display the policy configuration.</td><td><pre>vyatta@R1# <b>show policy</b>
 prefix-list ALLOW-PREFIXES {
     rule 1 {
         action permit
         prefix 12.0.0.0/8
     }
 }
 route-map eBGP-IMPORT {
     rule 10 {
         action permit
         match {
             ip {
                 address {
                     prefix-list ALLOW-PREFIXES
                 }
             }
         }
     }
     rule 20 {
         action deny
     }
 }
[edit]
vyatta@R1#</pre></td></tr>
<tr><td>R1</td><td>Display the BGP configuration for eBGP neighbor 88.88.88.2.</td><td><pre>vyatta@R1# <b>show protocols bgp 100 neighbor 88.88.88.2</b>
 remote-as 200
 route-map {
     import eBGP-IMPORT
 }
[edit]
vyatta@R1#</pre></td></tr>
</table>

Example 1-14 Creating an import policy

R4	Create a rule to match any prefix from 10.0.0.0/8 to 32.	vyatta@R4# **set policy prefix-list RFC1918PREFIXES rule 1 action permit** [edit] vyatta@R4# **set policy prefix-list RFC1918PREFIXES rule 1 le 32** [edit] vyatta@R4# **set policy prefix-list RFC1918PREFIXES rule 1 prefix 10.0.0.0/8** [edit]
R4	Create a rule to match any prefix from 172.16.0.0/12 to 32.	vyatta@R4# **set policy prefix-list RFC1918PREFIXES rule 2 action permit** [edit] vyatta@R4# **set policy prefix-list RFC1918PREFIXES rule 2 le 32** [edit] vyatta@R4# **set policy prefix-list RFC1918PREFIXES rule 2 prefix 172.16.0.0/12** [edit]
R4	Create a rule to match any prefix from 192.168.0.0/16 to 32.	vyatta@R4# **set policy prefix-list RFC1918PREFIXES rule 3 action permit** [edit] vyatta@R4# **set policy prefix-list RFC1918PREFIXES rule 3 le 32** [edit] vyatta@R4# **set policy prefix-list RFC1918PREFIXES rule 3 prefix 192.168.0.0/16** [edit]
R4	Create a route map rule to deny all prefixes in our list.	vyatta@R4# **set policy route-map eBGP-IMPORT rule 10 action deny** [edit] vyatta@R4# **set policy route-map eBGP-IMPORT rule 10 match ip address prefix-list RFC1918PREFIXES** [edit]
R4	Create a route map rule to permit all other prefixes.	vyatta@R4# **set policy route-map eBGP-IMPORT rule 20 action permit** [edit]

Example 1-14 Creating an import policy

R4	Assign the route map policy created as the import route map policy for AS 300.	vyatta@R4# **set protocols bgp 100 neighbor 99.99.99.2 route-map import eBGP-IMPORT** [edit]
R4	Commit the configuration.	vyatta@R4# **commit** [edit]
R4	Reset the BGP session to the peer so that the new policies are enabled.	vyatta@R1# **run clear ip bgp 99.99.99.2** [edit]

Example 1-14 Creating an import policy

R4 Display the policy configuration.

```
vyatta@R4# show policy
 prefix-list RFC1918PREFIXES {
     rule 1 {
         action permit
         le 32
         prefix 10.0.0.0/8
     }
     rule 2 {
         action permit
         le 32
         prefix 172.16.0.0/12
     }
     rule 3 {
         action permit
         le 32
         prefix 192.168.0.0/16
     }
 }
 route-map eBGP-IMPORT {
     rule 10 {
         action deny
         match {
             ip {
                 address {
                     prefix-list RFC1918PREFIXES
                 }
             }
         }
     }
     rule 20 {
         action permit
     }
 }
[edit]
vyatta@R4#
```

R4 Display the BGP configuration for eBGP neighbor 99.99.99.2.

```
vyatta@R4# show protocols bgp 100 neighbor 99.99.99.2
 remote-as 300
 route-map {
     import eBGP-IMPORT
 }
[edit]
vyatta@R4#
```

Verifying the Inbound Filter

The following commands can be used to verify the inbound filter configuration.

R1: show ip bgp

Example 1-15 shows R1's BGP table *before* the import filter is applied.

Example 1-15 R1 inbound BGP routes before import filtering

```
vyatta@R1:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 2.0.0.0/24       88.88.88.2               0              0 200 i
*> 2.1.0.0/24       88.88.88.2               0              0 200 i
*> 2.2.0.0/24       88.88.88.2               0              0 200 i
*>i3.0.0.0/24       99.99.99.2               0     100      0 300 i
*>i3.1.0.0/24       99.99.99.2               0     100      0 300 i
*>i3.2.0.0/24       99.99.99.2               0     100      0 300 i
*> 12.0.0.0         88.88.88.2               0              0 200 i
*>i13.0.0.0/24      99.99.99.2               0     100      0 300 i
*> 88.88.88.0/30    88.88.88.2               0              0 200 i
*>i99.99.99.0/30    99.99.99.2               0     100      0 300 i
*> 172.16.0.0/24    0.0.0.0                  1          32768 i
* i                 10.0.0.44                1     100      0 i
*>i172.16.128.0/24  99.99.99.2               0     100      0 300 i
*>i192.168.2.0      99.99.99.2               0     100      0 300 i

Total number of prefixes 13
vyatta@R1:~$
```

R1: show ip bgp

Example 1-16 shows R1's BGP table *after* the import filter is applied.

Example 1-16 R1 inbound BGP routes after import filtering

```
vyatta@R1:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i3.0.0.0/24       99.99.99.2               0    100      0 300 i
*>i3.1.0.0/24       99.99.99.2               0    100      0 300 i
*>i3.2.0.0/24       99.99.99.2               0    100      0 300 i
*> 12.0.0.0         88.88.88.2               0             0 200 i
*>i13.0.0.0/24      99.99.99.2               0    100      0 300 i
*>i99.99.99.0/30    99.99.99.2               0    100      0 300 i
*> 172.16.0.0/24    0.0.0.0                  1         32768 i
* i                 10.0.0.44                1    100      0 i
*>i172.16.128.0/24  99.99.99.2               0    100      0 300 i
*>i192.168.2.0      99.99.99.2               0    100      0 300 i

Total number of prefixes 9
vyatta@R1:~$
```

Note that only 12.0.0.0 from 88.88.88.2 is still in the table.

R4: show ip bgp

Example 1-17 shows R4’s BGP table *before* the import filter is applied.

Example 1-17 R4 inbound BGP routes before import filtering

```
vyatta@R4:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.44
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 3.0.0.0/24       99.99.99.2               0             0 300 i
*> 3.1.0.0/24       99.99.99.2               0             0 300 i
*> 3.2.0.0/24       99.99.99.2               0             0 300 i
*>i12.0.0.0         88.88.88.2               0    100      0 200 i
*> 13.0.0.0/24      99.99.99.2               0             0 300 i
*> 99.99.99.0/30    99.99.99.2               0             0 300 i
* i172.16.0.0/24    10.0.0.11                1    100      0 i
*>                  0.0.0.0                  1         32768 i
*> 172.16.128.0/24  99.99.99.2               0             0 300 i
*> 192.168.2.0      99.99.99.2               0             0 300 i

Total number of prefixes 9
```

```
vyatta@R4:~$
```

R4: show ip bgp

The output below shows R4’s BGP table *after* the import filter is applied.

Example 1-18 R4 inbound BGP routes after import filtering

```
vyatta@R4:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.44
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 3.0.0.0/24       99.99.99.2               0             0 300 i
*> 3.1.0.0/24       99.99.99.2               0             0 300 i
*> 3.2.0.0/24       99.99.99.2               0             0 300 i
*>i12.0.0.0         88.88.88.2               0    100      0 200 i
*> 13.0.0.0/24      99.99.99.2               0             0 300 i
*> 99.99.99.0/30    99.99.99.2               0             0 300 i
* i172.16.0.0/24    10.0.0.11                1    100      0 i
*>                  0.0.0.0                  1         32768 i

Total number of prefixes 7
vyatta@R4:~$
```

Outbound Route Filtering

Filtering outbound prefixes is another common BGP configuration requirement. On the Vyatta system this is accomplished using routing policies that are then applied to the BGP process as “export” policies.

The example in this section assumes that AS100 does not want to be a transit AS for AS 200 or AS 300. This means that:

- eBGP routes from R1’s eBGP peer (AS 200) should not be sent to R4’s eBGP peer.
- Routes from R4’s eBGP peer (AS 300) should not be sent to R1’s eBGP peer.

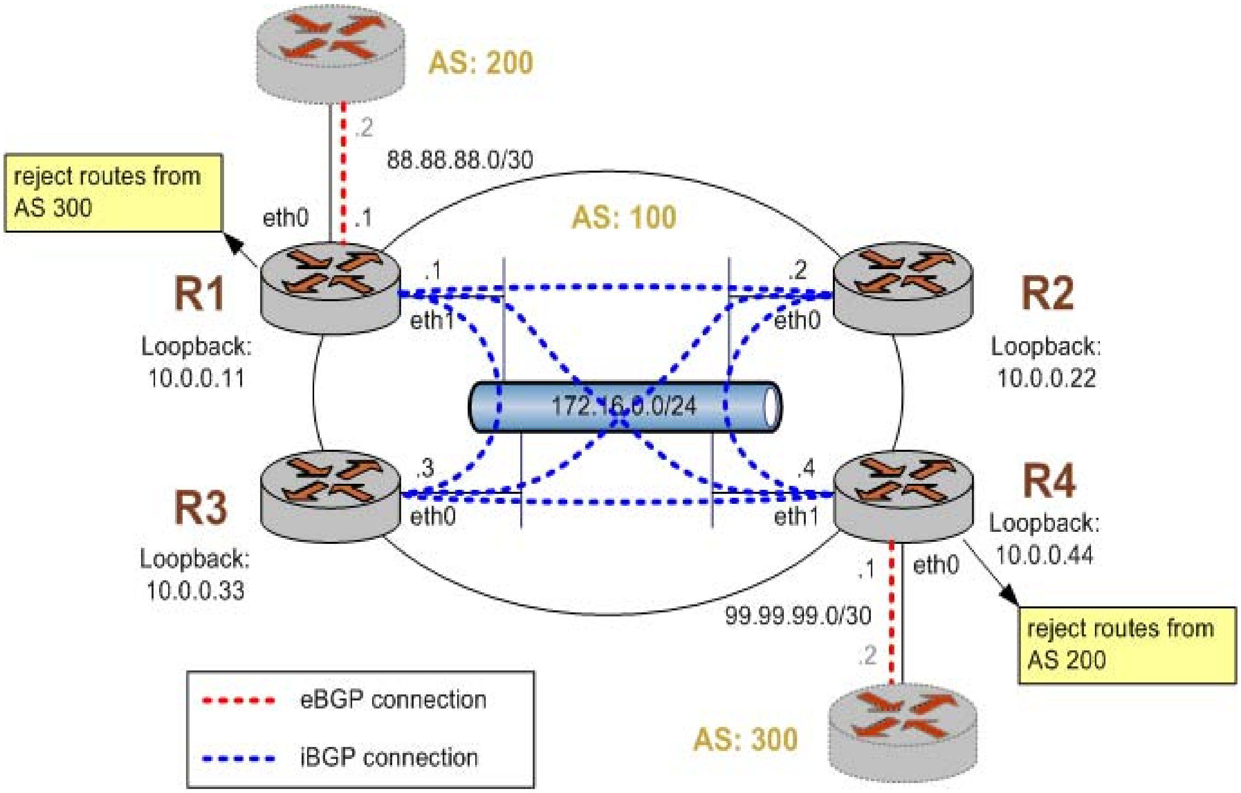
If we *did not* implement this filtering, AS 300 might send traffic destined for AS 200 to router R4, and this traffic would then be carried across the AS 100 network.

There are several ways that this routing policy could be implemented: two most common are basing the filter on the network prefix or basing it on the AS Path. In this example, we update the existing BGP export policy to add some additional restrictions that will prevent AS 100 from acting as a transit network for AS 200 and AS 300.

This export policy is shown in Figure 1-10.

NOTE *We assume that the routers in AS200 and AS300 are configured appropriately as eBGP peers.*

Figure 1-10 Filtering outbound routes



To create this export policy, perform the following steps in configuration mode:

Example 1-19 Creating an export policy

Router	Step	Command(s)
R1	Create a list of AS paths to deny. In this case we just have one - AS300.	vyatta@R1# **set policy as-path-list AS300 rule 1 action permit** [edit] vyatta@R1# **set policy as-path-list AS300 rule 1 regex 300** [edit]

Example 1-19 Creating an export policy

R1	Create a route map rule to deny all AS paths in our list.	vyatta@R1# **set policy route-map eBGP-EXPORT rule 10 action deny** [edit] vyatta@R1# **set policy route-map eBGP-EXPORT rule 10 match as-path AS300** [edit]
R1	Create a route map rule to permit all other prefixes.	vyatta@R1# **set policy route-map eBGP-IMPORT rule 20 action permit** [edit]
R1	Assign the route map policy created as the export route map policy for AS 200.	vyatta@R1# **set protocols bgp 100 neighbor 88.88.88.2 route-map export eBGP-EXPORT** [edit]
R1	Commit the configuration.	vyatta@R1# **commit** [edit]
R1	Reset the BGP session to the peer so that the new policies are enabled.	vyatta@R1# **run clear ip bgp 88.88.88.2** [edit]

R1	Display the policy configurations.	vyatta@R1# **show policy as-path-list AS300**
		rule 1 {
		action permit
		regex 300
		}
		[edit]
		vyatta@R1# **show policy route-map eBGP-EXPORT**
		rule 10 {
		action deny
		match {
		as-path AS300
		}
		}
		rule 20 {
		action permit
		}
		[edit]
		vyatta@R1#

Example 1-19 Creating an export policy

R1	Display the BGP configuration for eBGP neighbor 88.88.88.2.	vyatta@R1# **show protocols bgp 100 neighbor 88.88.88.2**
		 remote-as 200
		 route-map {
		     export eBGP-EXPORT
		     import eBGP-IMPORT
		 }
		[edit]
		vyatta@R1#

R4	Create a list of AS paths to deny. In this case we just have one - AS200.	vyatta@R4# **set policy as-path-list AS200 rule 1 action permit** [edit] vyatta@R4# **set policy as-path-list AS200 rule 1 regex 200** [edit]
R4	Create a route map rule to deny all AS paths in our list.	vyatta@R4# **set policy route-map eBGP-EXPORT rule 10 action deny** [edit] vyatta@R4# **set policy route-map eBGP-EXPORT rule 10 match as-path AS200** [edit]
R4	Create a route map rule to permit all other prefixes.	vyatta@R4# **set policy route-map eBGP-IMPORT rule 20 action permit** [edit]
R4	Assign the route map policy created as the export route map policy for AS 300.	vyatta@R4# **set protocols bgp 100 neighbor 99.99.99.2 route-map export eBGP-EXPORT** [edit]
R4	Commit the configuration.	vyatta@R4# **commit** [edit]
R4	Reset the BGP session to the peer so that the new policies are enabled.	vyatta@R4# **run clear ip bgp 99.99.99.2** [edit]

Example 1-19 Creating an export policy

R4	Display the policy configurations.	`vyatta@R4# show policy as-path-list AS200`
		` rule 1 {`
		`     action permit`
		`     regex 200`
		` }`
		`[edit]`
		`vyatta@R4# show policy route-map eBGP-EXPORT`
		` rule 10 {`
		`     action deny`
		`     match {`
		`         as-path AS200`
		`     }`
		` }`
		` rule 20 {`
		`     action permit`
		` }`
		`[edit]`
		`vyatta@R4#`

R4	Display the BGP configuration for eBGP neighbor 99.99.99.2.	`vyatta@R4# show protocols bgp 100 neighbor 99.99.99.2`
		` remote-as 300`
		` route-map {`
		`     export eBGP-EXPORT`
		`     import eBGP-IMPORT`
		` }`
		`[edit]`
		`vyatta@R4#`

Verifying the Outbound Filter

The following commands can be used to verify the outbound filter configuration.

AS 200: show ip bgp

Example 1-20 shows AS 200’s BGP table *before* the export filter is applied.

Example 1-20 AS 200 outbound BGP routes before export filtering

```
vyatta@AS200:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
   Network          Next Hop            Metric LocPrf Weight Path
*> 2.0.0.0/24       0.0.0.0                  0          32768 i
*> 2.1.0.0/24       0.0.0.0                  0          32768 i
*> 2.2.0.0/24       0.0.0.0                  0          32768 i
*> 3.0.0.0/24       88.88.88.1                              0 100 300 i
*> 3.1.0.0/24       88.88.88.1                              0 100 300 i
*> 3.2.0.0/24       88.88.88.1                              0 100 300 i
*> 12.0.0.0         0.0.0.0                  0          32768 i
*> 13.0.0.0/24      88.88.88.1                              0 100 300 i
*> 88.88.88.0/30    0.0.0.0                  0          32768 i
*> 99.99.99.0/30    88.88.88.1                              0 100 300 i
*> 172.16.0.0/24    88.88.88.1               1              0 100 i

Total number of prefixes 11
vyatta@AS200:~$
```

AS 200: show ip bgp

Example 1-21 shows AS 200’s BGP table *after* the export filter is applied.

Example 1-21 AS 200 outbound BGP routes after export filtering

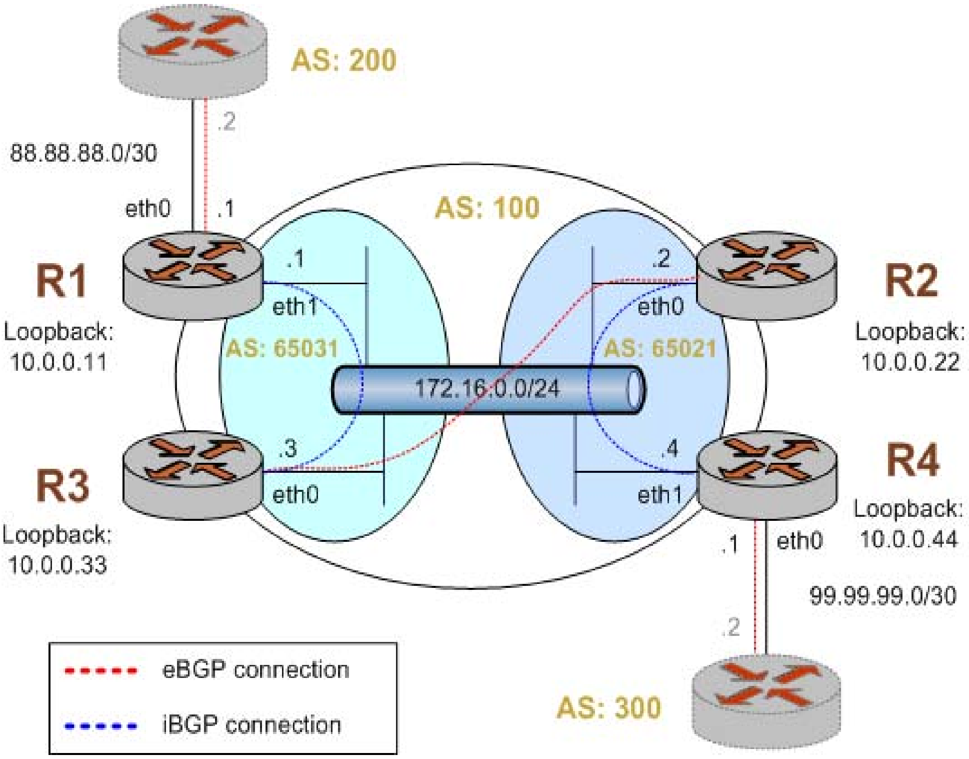
```
vyatta@AS200:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 2.0.0.0/24       0.0.0.0                  0          32768 i
*> 2.1.0.0/24       0.0.0.0                  0          32768 i
*> 2.2.0.0/24       0.0.0.0                  0          32768 i
*> 12.0.0.0         0.0.0.0                  0          32768 i
*> 88.88.88.0/30    0.0.0.0                  0          32768 i
*> 172.16.0.0/24    88.88.88.1               1              0 100 i

Total number of prefixes 6
vyatta@AS200:~$
```

Confederations

Confederations allow large Autonomous Systems to sub-divide the AS into sub-ASs. This helps solve the scalability issues associated with having to maintain a full mesh of iBGP connections between all iBGP routers in the AS. In the confederation example shown in Figure 1-11, routers R1 and R3 are configured in one sub-AS (AS number 65031) and routers R2 and R4 are configured in a different sub-AS (AS number 65021).

NOTE *We assume that the routers in AS200 and AS300 are configured appropriately as eBGP peers.*

Figure 1-11 BGP confederation



This example assumes that the configurations in previous sections have been performed.

To create the confederation shown in Figure 1-11, perform the following steps in configuration mode

Example 1-22 Creating a BGP confederation

Router	Step	Command(s)

R1	Delete current BGP configuration.	`vyatta@R1#` **`delete protocols bgp 100`**
		`[edit]`

R1	To allow routes from AS200 to be injected into RIB on R3.	`vyatta@R1#` **`set protocols bgp 65031 neighbor 10.0.0.33 nexthop-self`**
		`[edit]`

R1	Set R3 in the same sub-AS as R1.	`vyatta@R1#` **`set protocols bgp 65031 neighbor 10.0.0.33 remote-as 65031`**
		`[edit]`

R1	Define the IP address on the local R1 router that is used to peer with the R3 router.	`vyatta@R1#` **`set protocols bgp 65031 neighbor 10.0.0.33 update-source 10.0.0.11`**
		`[edit]`

R1	Set the AS200 router in a different AS than R1.	`vyatta@R1#` **`set protocols bgp 65031 neighbor 88.88.88.2 remote-as 200`**
		`[edit]`

R1	Set the outbound filter.	`vyatta@R1#` **`set protocols bgp 65031 neighbor 88.88.88.2 route-map export eBGP-EXPORT`**
		`[edit]`

R1	Set the inbound filter.	`vyatta@R1#` **`set protocols bgp 65031 neighbor 88.88.88.2 route-map import eBGP-IMPORT`**
		`[edit]`

R1	Set the network to advertise.	`vyatta@R1#` **`set protocols bgp 65031 network 172.16.0.0/24`**
		`[edit]`

R1	Set the AS identifier for the confederation.	`vyatta@R1#` **`set protocols bgp 65031 parameters confederation identifier 100`**
		`[edit]`

R1	Set the peer for the sub-AS.	`vyatta@R1#` **`set protocols bgp 65031 parameters confederation peers 65021`**
		`[edit]`

Example 1-22 Creating a BGP confederation

R1	Set the router ID.	`vyatta@R1#` **`set protocols bgp 65031 parameters router-id 10.0.0.11`**
		`[edit]`

R1	Commit the configuration.	`vyatta@R1#` **`commit`**
		`[edit]`

R1	Display the BGP configuration.	`vyatta@R1#` **`show protocols bgp`**
		` 65031 {`
		`     neighbor 10.0.0.33 {`
		`         nexthop-self`
		`         remote-as 65031`
		`         update-source 10.0.0.11`
		`     }`
		`     neighbor 88.88.88.2 {`
		`         remote-as 200`
		`         route-map {`
		`             export eBGP-EXPORT`
		`             import eBGP-IMPORT`
		`         }`
		`     }`
		`     network 172.16.0.0/24 {`
		`     }`
		`     parameters {`
		`         confederation {`
		`             identifier 100`
		`             peers 65021`
		`         }`
		`         router-id 10.0.0.11`
		`     }`
		` }`
		`[edit]`
		`vyatta@R1#`

R2	Delete current BGP configuration.	`vyatta@R2#` **`delete protocols bgp 100`**
		`[edit]`

R2	Set R3 in a different sub-AS than R2.	`vyatta@R2#` **`set protocols bgp 65021 neighbor 10.0.0.33 remote-as 65031`**
		`[edit]`

Example 1-22 Creating a BGP confederation

R2	Define the IP address on the local R2 router that is used to peer with the R3 router.	vyatta@R2# **set protocols bgp 65021 neighbor 10.0.0.33 update-source 10.0.0.22** [edit]
R2	Set R4 in the same sub-AS as R2.	vyatta@R2# **set protocols bgp 65021 neighbor 10.0.0.44 remote-as 65021** [edit]
R2	Define the IP address on the local R2 router that is used to peer with the R4 router.	vyatta@R2# **set protocols bgp 65021 neighbor 10.0.0.44 update-source 10.0.0.22** [edit]
R2	Set the network to advertise.	vyatta@R2# **set protocols bgp 65021 network 172.16.0.0/24** [edit]
R2	Set the AS identifier for the confederation.	vyatta@R2# **set protocols bgp 65021 parameters confederation identifier 100** [edit]
R2	Set the peer for the sub-AS.	vyatta@R2# **set protocols bgp 65021 parameters confederation peers 65031** [edit]
R2	Set the router ID.	vyatta@R2# **set protocols bgp 65021 parameters router-id 10.0.0.22** [edit]
R2	Commit the configuration.	vyatta@R2# **commit** [edit]

Example 1-22 Creating a BGP confederation

R2	Display the BGP configuration.	`vyatta@R2#` **`show protocols bgp`**
		` 65021 {`
		`     neighbor 10.0.0.33 {`
		`         remote-as 65031`
		`         update-source 10.0.0.22`
		`     }`
		`     neighbor 10.0.0.44 {`
		`         remote-as 65021`
		`         update-source 10.0.0.22`
		`     }`
		`     network 172.16.0.0/24 {`
		`     }`
		`     parameters {`
		`         confederation {`
		`             identifier 100`
		`             peers 65031`
		`         }`
		`         router-id 10.0.0.22`
		`     }`
		` }`
		`[edit]`
		`vyatta@R2#`

R3	Delete current BGP configuration.	`vyatta@R3#` **`delete protocols bgp 100`**
		`[edit]`

R3	Set R1 in the same sub-AS as R3.	`vyatta@R3#` **`set protocols bgp 65031 neighbor 10.0.0.11 remote-as 65031`**
		`[edit]`

R3	Define the IP address on the local R3 router that is used to peer with the R1 router.	`vyatta@R3#` **`set protocols bgp 65031 neighbor 10.0.0.11 update-source 10.0.0.33`**
		`[edit]`

R3	Set the R2 router in a different AS than R3.	`vyatta@R3#` **`set protocols bgp 65031 neighbor 10.0.0.22 remote-as 65021`**
		`[edit]`

Example 1-22 Creating a BGP confederation

<table>
<tr><td>R3</td><td>Define the IP address on the local R3 router that is used to peer with the R2 router.</td><td><pre>vyatta@R3# set protocols bgp 65031 neighbor 10.0.0.22
update-source 10.0.0.33
[edit]</pre></td></tr>
<tr><td>R3</td><td>Set the network to advertise.</td><td><pre>vyatta@R3# set protocols bgp 65031 network 172.16.0.0/24
[edit]</pre></td></tr>
<tr><td>R3</td><td>Set the AS identifier for the confederation.</td><td><pre>vyatta@R3# set protocols bgp 65031 parameters confederation
identifier 100
[edit]</pre></td></tr>
<tr><td>R3</td><td>Set the peer for the sub-AS.</td><td><pre>vyatta@R3# set protocols bgp 65031 parameters confederation
peers 65021
[edit]</pre></td></tr>
<tr><td>R3</td><td>Set the router ID.</td><td><pre>vyatta@R3# set protocols bgp 65031 parameters router-id
10.0.0.33
[edit]</pre></td></tr>
<tr><td>R3</td><td>Commit the configuration.</td><td><pre>vyatta@R3# commit
[edit]</pre></td></tr>
<tr><td>R3</td><td>Display the BGP configuration.</td><td><pre>vyatta@R3# show protocols bgp
 65031 {
     neighbor 10.0.0.11 {
         remote-as 65031
         update-source 10.0.0.33
     }
     neighbor 10.0.0.22 {
         remote-as 65021
         update-source 10.0.0.33
     }
     network 172.16.0.0/24 {
     }
     parameters {
         confederation {
             identifier 100
             peers 65021
         }
         router-id 10.0.0.33
     }
 }
[edit]
vyatta@R3#</pre></td></tr>
</table>

Example 1-22 Creating a BGP confederation

R4	Delete current BGP configuration.	vyatta@R4# **delete protocols bgp 100** [edit]
R4	To allow routes from AS300 to be injected into RIB on R2.	vyatta@R4# **set protocols bgp 65021 neighbor 10.0.0.22 nexthop-self** [edit]
R4	Set R2 in the same sub-AS as R4.	vyatta@R4# **set protocols bgp 65021 neighbor 10.0.0.22 remote-as 65021** [edit]
R4	Define the IP address on the local R4 router that is used to peer with the R2 router.	vyatta@R4# **set protocols bgp 65021 neighbor 10.0.0.22 update-source 10.0.0.44** [edit]
R4	Set the AS300 router in a different AS than R4.	vyatta@R4# **set protocols bgp 65021 neighbor 99.99.99.2 remote-as 300** [edit]
R4	Set the outbound filter.	vyatta@R4# **set protocols bgp 65021 neighbor 99.99.99.2 route-map export eBGP-EXPORT** [edit]
R4	Set the inbound filter.	vyatta@R4# **set protocols bgp 65021 neighbor 99.99.99.2 route-map import eBGP-IMPORT** [edit]
R4	Set the network to advertise.	vyatta@R4# **set protocols bgp 65021 network 172.16.0.0/24** [edit]
R4	Set the AS identifier for the confederation.	vyatta@R4# **set protocols bgp 65021 parameters confederation identifier 100** [edit]
R4	Set the peer for the sub-AS.	vyatta@R4# **set protocols bgp 65021 parameters confederation peers 65031** [edit]
R4	Set the router ID.	vyatta@R4# **set protocols bgp 65021 parameters router-id 10.0.0.44** [edit]

Example 1-22 Creating a BGP confederation

<table>
<tr><td>R4</td><td>Commit the configuration.</td><td><pre>vyatta@R4# commit
[edit]</pre></td></tr>
<tr><td>R4</td><td>Display the BGP configuration.</td><td><pre>vyatta@R4# show protocols bgp
 65021 {
     neighbor 10.0.0.22 {
         nexthop-self
         remote-as 65021
         update-source 10.0.0.44
     }
     neighbor 99.99.99.2 {
         remote-as 300
         route-map {
             export eBGP-EXPORT
             import eBGP-IMPORT
         }
     }
     network 172.16.0.0/24 {
     }
     parameters {
         confederation {
             identifier 100
             peers 65031
         }
         router-id 10.0.0.44
     }
 }
[edit]
vyatta@R4#</pre></td></tr>
</table>

Verifying the Confederation

The following commands can be used to verify the confederation configuration.

R1: show ip bgp summary

Example 1-23 shows the output of the **show ip bgp summary** command for router R1 at this stage of the configuration.

Example 1-23 Verifying confederations on R1: “show ip bgp summary”

```
vyatta@R1:~$ show ip bgp summary
BGP router identifier 10.0.0.11, local AS number 65031
```

```
RIB entries 13, using 832 bytes of memory
Peers 2, using 5040 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.33       4 65031    1159    1167        0    0    0 00:00:23        6
88.88.88.2      4   200    1150    1159        0    0    0 00:00:06        1

Total number of neighbors 2
vyatta@R1:~$
```

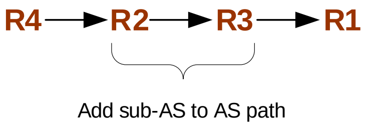
R1: show ip bgp

Example 1-24 shows the output of the **show ip bgp** command for router R1 at this stage of the configuration.

Example 1-24 Verifying confederations on R1: “show ip bgp”

```
vyatta@R1:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i3.0.0.0/24       10.0.0.44                0    100      0 (65021) 300 i
*>i3.1.0.0/24       10.0.0.44                0    100      0 (65021) 300 i
*>i3.2.0.0/24       10.0.0.44                0    100      0 (65021) 300 i
*> 12.0.0.0         88.88.88.2               0             0 200 i
*>i13.0.0.0/24      10.0.0.44                0    100      0 (65021) 300 i
*>i99.99.99.0/30    10.0.0.44                0    100      0 (65021) 300 i
* i172.16.0.0/24    10.0.0.33                1    100      0 i
*>                  0.0.0.0                  1         32768 i

Total number of prefixes 7
vyatta@R1:~$
```

Note that the routes learned from router R4 (Next Hop 10.0.0.44) include the confederation sub-AS in the AS Path. All confederation sub-ASs will be shown inside brackets (). This information is not transmitted outside of the true AS (AS 100).



R2: show ip bgp summary

Example 1-25 shows the output of the **show ip bgp summary** command for router R2 at this stage of the configuration.

Example 1-25 Verifying confederations on R2: “show ip bgp summary”

```
vyatta@R2:~$ show ip bgp summary
BGP router identifier 10.0.0.22, local AS number 65021
RIB entries 13, using 832 bytes of memory
Peers 2, using 5040 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.33       4 65031    1165    1163        0    0    0 00:23:14         2
10.0.0.44       4 65021    1159    1167        0    0    0 00:23:32         6

Total number of neighbors 2
vyatta@R2:~$
```

R2: show ip bgp

Example 1-26 shows the output of the **show ip bgp** command for router R2 at this stage of the configuration.

Example 1-26 Verifying confederations on R2: “show ip bgp”

```
vyatta@R2:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.22
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i3.0.0.0/24       10.0.0.44                0    100      0 300 i
*>i3.1.0.0/24       10.0.0.44                0    100      0 300 i
*>i3.2.0.0/24       10.0.0.44                0    100      0 300 i
*> 12.0.0.0         10.0.0.11                0    100      0 (65031) 200 i
```

```
*>i13.0.0.0/24      10.0.0.44                0    100      0 300 i
*>i99.99.99.0/30    10.0.0.44                0    100      0 300 i
*  172.16.0.0/24    10.0.0.33                1    100      0 (65031) i
* i                 10.0.0.44                1    100      0 i
*>                  0.0.0.0                  1         32768 i

Total number of prefixes 7
vyatta@R2:~$
```

R3: show ip bgp summary

Example 1-27 shows the output of the **show bgp peers** command for router R3 at this stage of the configuration.

Example 1-27 Verifying confederations on R3: “show bgp peers”

```
vyatta@R3:~$ show ip bgp summary
BGP router identifier 10.0.0.33, local AS number 65031
RIB entries 13, using 832 bytes of memory
Peers 2, using 5040 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.11       4 65031    1172    1180        0    0    0 00:20:04         2
10.0.0.22       4 65021    1161    1177        0    0    0 00:27:51         6

Total number of neighbors 2
vyatta@R3:~$
```

R3: show ip bgp

Example 1-28 shows the output of the **show ip bgp** command for router R3 at this stage of the configuration.

Example 1-28 Verifying confederations on R3: “show ip bgp”

```
vyatta@R3:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.33
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
```

```
*> 3.0.0.0/24       10.0.0.44                0    100      0 (65021) 300 i
*> 3.1.0.0/24       10.0.0.44                0    100      0 (65021) 300 i
*> 3.2.0.0/24       10.0.0.44                0    100      0 (65021) 300 i
*>i12.0.0.0         10.0.0.11                0    100      0 200 i
*> 13.0.0.0/24      10.0.0.44                0    100      0 (65021) 300 i
*> 99.99.99.0/30    10.0.0.44                0    100      0 (65021) 300 i
* i172.16.0.0/24    10.0.0.11                1    100      0 i
*                   10.0.0.22                1    100      0 (65021) i
*>                  0.0.0.0                  1         32768 i

Total number of prefixes 7
vyatta@R3:~$
```

R4: show ip bgp summary

Example 1-29 shows the output of the **show ip bgp summary** command for router R4 at this stage of the configuration.

Example 1-29 Verifying confederations on R4: “show ip bgp summary”

```
vyatta@R4:~$ show ip bgp summary
BGP router identifier 10.0.0.44, local AS number 65021
RIB entries 13, using 832 bytes of memory
Peers 2, using 5040 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.22       4 65021    1165    1168        0    0    0 00:32:56        2
99.99.99.2      4   300    1155    1162        0    0    0 00:33:30        5

Total number of neighbors 2
vyatta@R4:~$
```

R4: show ip bgp

Example 1-30 shows the output of the **show ip bgp** command for router R4 at this stage of the configuration.

Example 1-30 Verifying confederations on R4: “show ip bgp”

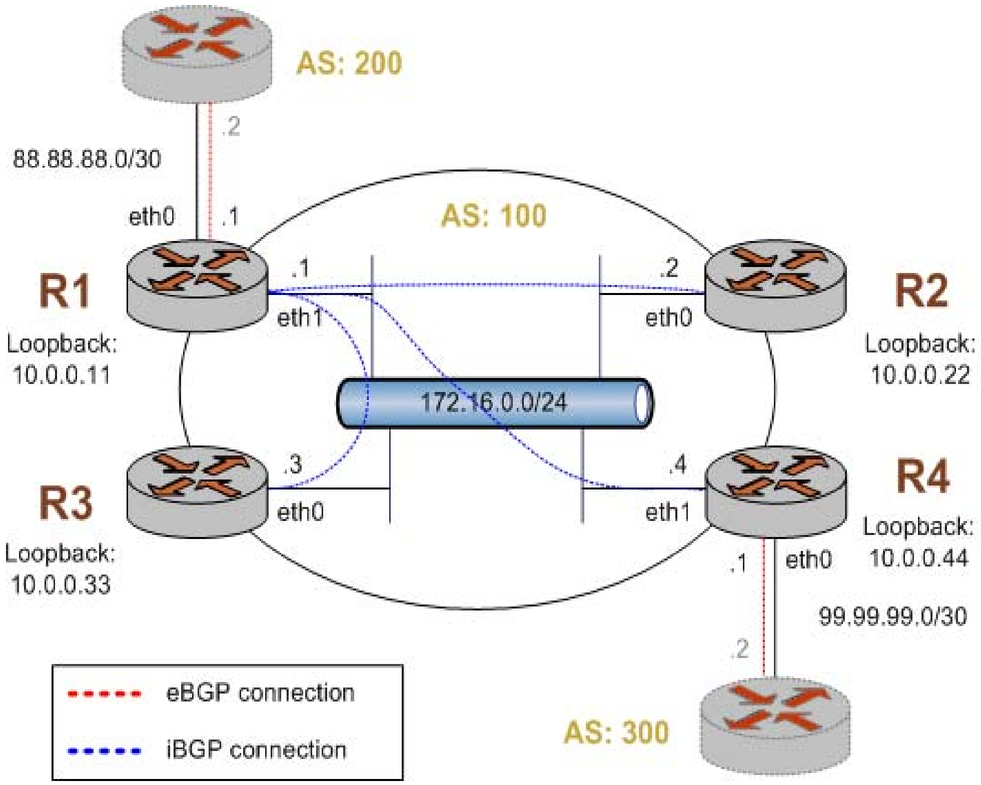
```
vyatta@R4:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.44
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 3.0.0.0/24       99.99.99.2               0             0 300 i
*> 3.1.0.0/24       99.99.99.2               0             0 300 i
*> 3.2.0.0/24       99.99.99.2               0             0 300 i
*>i12.0.0.0         10.0.0.11                0    100      0 (65031) 200 i
*> 13.0.0.0/24      99.99.99.2               0             0 300 i
*> 99.99.99.0/30    99.99.99.2               0             0 300 i
* i172.16.0.0/24    10.0.0.22                1    100      0 i
*>                  0.0.0.0                  1         32768 i

Total number of prefixes 7
vyatta@R4:~$
```

Route Reflectors

Router reflectors are another technology designed to help BGP scale to large Autonomous Systems. In a route reflector configuration there is at least one route reflector server and one or more route reflector clients. In the example shown in Figure 1-12, router R1 is the route reflector server and router R2, R3, and R4 are the route reflector clients.

NOTE *We assume that the routers in AS200 and AS300 are configured appropriately as eBGP peers.*

Figure 1-12 BGP route reflector



This example assumes that the configurations in previous sections have been performed, and that interfaces and OSPF have been configured. If you are starting from a clean base system you need not delete previous configuration.

Example 1-31 Creating route reflectors

Router	Step	Command(s)

R1	Delete current BGP configuration.	`vyatta@R1# delete protocols bgp`
		`[edit]`

R1	To allow routes from AS200 to be injected into RIB on R2.	`vyatta@R1# set protocols bgp 100 neighbor 10.0.0.22 nexthop-self`
		`[edit]`

R1	Set R2 in the same AS as R1.	`vyatta@R1# set protocols bgp 100 neighbor 10.0.0.22 remote-as 100`
		`[edit]`

R1	Set R2 as a route reflector client.	`vyatta@R1# set protocols bgp 100 neighbor 10.0.0.22 route-reflector-client`
		`[edit]`

Example 1-31 Creating route reflectors

R1	Define the IP address on the local R1 router that is used to peer with the R2 router.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.22 update-source 10.0.0.11** [edit]
R1	To allow routes from AS200 to be injected into RIB on R3.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.33 nexthop-self** [edit]
R1	Set R3 in the same AS as R1.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.33 remote-as 100** [edit]
R1	Set R3 as a route reflector client.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.33 route-reflector-client** [edit]
R1	Define the IP address on the local R1 router that is used to peer with the R3 router.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.33 update-source 10.0.0.11** [edit]
R1	To allow routes from AS200 to be injected into RIB on R4.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.44 nexthop-self** [edit]
R1	Set R4 in the same AS as R1.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.44 remote-as 100** [edit]
R1	Set R4 as a route reflector client.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.44 route-reflector-client** [edit]
R1	Define the IP address on the local R1 router that is used to peer with the R4 router.	vyatta@R1# **set protocols bgp 100 neighbor 10.0.0.44 update-source 10.0.0.11** [edit]
R1	Set the AS200 router in a different AS than R1.	vyatta@R1# **set protocols bgp 100 neighbor 88.88.88.2 remote-as 200** [edit]
R1	Set the outbound filter.	vyatta@R1# **set protocols bgp 100 neighbor 88.88.88.2 route-map export eBGP-EXPORT** [edit]

Example 1-31 Creating route reflectors

R1	Set the inbound filter.	`vyatta@R1#` **`set protocols bgp 100 neighbor 88.88.88.2 route-map import eBGP-IMPORT`**
		`[edit]`

R1	Set the network to advertise.	`vyatta@R1#` **`set protocols bgp 100 network 172.16.0.0/24`**
		`[edit]`

R1	Set the router ID.	`vyatta@R1#` **`set protocols bgp 100 parameters router-id 10.0.0.11`**
		`[edit]`

R1	Commit the configuration.	`vyatta@R1#` **`commit`**
		`[edit]`

Example 1-31 Creating route reflectors

R1	Display the BGP configuration.	vyatta@R1# **show protocols bgp**
		 100 {
		     neighbor 10.0.0.22 {
		         nexthop-self
		         remote-as 100
		         route-reflector-client
		         update-source 10.0.0.11
		     }
		     neighbor 10.0.0.33 {
		         nexthop-self
		         remote-as 100
		         route-reflector-client
		         update-source 10.0.0.11
		     }
		     neighbor 10.0.0.44 {
		         nexthop-self
		         remote-as 100
		         route-reflector-client
		         update-source 10.0.0.11
		     }
		     neighbor 88.88.88.2 {
		         remote-as 200
		         route-map {
		             export eBGP-EXPORT
		             import eBGP-IMPORT
		         }
		     }
		     network 172.16.0.0/24 {
		     }
		     parameters {
		         router-id 10.0.0.11
		     }
		 }
		[edit]
		vyatta@R1#

R2	Delete current BGP configuration.	vyatta@R2# **delete protocols bgp** [edit]
R2	Set R1 in the same AS as R2.	vyatta@R2# **set protocols bgp 100 neighbor 10.0.0.11 remote-as 100** [edit]

Example 1-31 Creating route reflectors

R2	Define the IP address on the local R2 router that is used to peer with the R1 router.	vyatta@R2# **set protocols bgp 100 neighbor 10.0.0.11 update-source 10.0.0.22** [edit]
R2	Set the network to advertise.	vyatta@R2# **set protocols bgp 100 network 172.16.0.0/24** [edit]
R2	Set the router ID.	vyatta@R2# **set protocols bgp 100 parameters router-id 10.0.0.22** [edit]
R2	Commit the configuration.	vyatta@R2# **commit** [edit]

R2	Display the BGP configuration.	vyatta@R2# **show protocols bgp**
		` 100 {`
		`     neighbor 10.0.0.11 {`
		`         remote-as 100`
		`         update-source 10.0.0.22`
		`     }`
		`     network 172.16.0.0/24 {`
		`     }`
		`     parameters {`
		`         router-id 10.0.0.22`
		`     }`
		` }`
		[edit]
		vyatta@R2#

R3	Delete current BGP configuration.	vyatta@R3# **delete protocols bgp** [edit]
R3	Set R1 in the same AS as R3.	vyatta@R3# **set protocols bgp 100 neighbor 10.0.0.11 remote-as 100** [edit]
R3	Define the IP address on the local R3 router that is used to peer with the R1 router.	vyatta@R3# **set protocols bgp 100 neighbor 10.0.0.11 update-source 10.0.0.33** [edit]
R3	Set the network to advertise.	vyatta@R3# **set protocols bgp 100 network 172.16.0.0/24** [edit]

Example 1-31 Creating route reflectors

R3	Set the router ID.	`vyatta@R3# `**`set protocols bgp 100 parameters router-id 10.0.0.33`**
		`[edit]`

R3	Commit the configuration.	`vyatta@R3# `**`commit`**
		`[edit]`

R3	Display the BGP configuration.	`vyatta@R3# `**`show protocols bgp`**
		` 100 {`
		`     neighbor 10.0.0.11 {`
		`         remote-as 100`
		`         update-source 10.0.0.33`
		`     }`
		`     network 172.16.0.0/24 {`
		`     }`
		`     parameters {`
		`         router-id 10.0.0.33`
		`     }`
		` }`
		`[edit]`
		`vyatta@R3#`

R4	Delete current BGP configuration.	`vyatta@R4# `**`delete protocols bgp`**
		`[edit]`

R4	To allow routes from AS300 to be injected into RIB on R1.	`vyatta@R4# `**`set protocols bgp 100 neighbor 10.0.0.11 nexthop-self`**
		`[edit]`

R4	Set R1 in the same AS as R4.	`vyatta@R4# `**`set protocols bgp 100 neighbor 10.0.0.11 remote-as 100`**
		`[edit]`

R4	Define the IP address on the local R4 router that is used to peer with the R1 router.	`vyatta@R4# `**`set protocols bgp 100 neighbor 10.0.0.11 update-source 10.0.0.44`**
		`[edit]`

R4	Set the AS300 router in a different AS than R4.	`vyatta@R4# `**`set protocols bgp 100 neighbor 99.99.99.2 remote-as 300`**
		`[edit]`

R4	Set the outbound filter.	`vyatta@R4# `**`set protocols bgp 100 neighbor 99.99.99.2 route-map export eBGP-EXPORT`**
		`[edit]`

Example 1-31 Creating route reflectors

<table>
<tr><td>R4</td><td>Set the inbound filter.</td><td><pre>vyatta@R4# set protocols bgp 100 neighbor 99.99.99.2
route-map import eBGP-IMPORT
[edit]</pre></td></tr>
<tr><td>R4</td><td>Set the network to advertise.</td><td><pre>vyatta@R4# set protocols bgp 100 network 172.16.0.0/24
[edit]</pre></td></tr>
<tr><td>R4</td><td>Set the router ID.</td><td><pre>vyatta@R4# set protocols bgp 100 parameters router-id
10.0.0.44
[edit]</pre></td></tr>
<tr><td>R4</td><td>Commit the configuration.</td><td><pre>vyatta@R4# commit
[edit]</pre></td></tr>
<tr><td>R4</td><td>Display the BGP configuration.</td><td><pre>vyatta@R4# show protocols bgp
 100 {
     neighbor 10.0.0.11 {
         nexthop-self
         remote-as 100
         update-source 10.0.0.44
     }
     neighbor 99.99.99.2 {
         remote-as 300
         route-map {
             export eBGP-EXPORT
             import eBGP-IMPORT
         }
     }
     network 172.16.0.0/24 {
     }
     parameters {
         router-id 10.0.0.44
     }
 }
[edit]
vyatta@R4#</pre></td></tr>
</table>

Verifying the Route Reflector

The following commands can be used to verify the route reflector configuration.

R1: show ip bgp summary

Example 1-32 shows the output of the **show ip bgp summary** command for router R1 at this stage of the configuration.

Example 1-32 Verifying route reflector on R1: “show ip bgp summary”

```
vyatta@R1:~$ show ip bgp summary
BGP router identifier 10.0.0.11, local AS number 100
RIB entries 13, using 832 bytes of memory
Peers 4, using 10080 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.22       4   100      46      66        0    0    0 00:26:12        1
10.0.0.33       4   100      48      66        0    0    0 00:25:50        1
10.0.0.44       4   100      40      54        0    0    0 00:25:05        6
88.88.88.2      4   200      33      36        0    0    0 00:27:14        1

Total number of neighbors 4
vyatta@R1:~$
```

R1: show ip bgp

Example 1-33 shows the output of the **show ip bgp** command for router R1 at this stage of the configuration.

Example 1-33 Verifying route reflector on R1: “show ip bgp”

```
vyatta@R1:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i3.0.0.0/24       10.0.0.44                0    100      0 300 i
*>i3.1.0.0/24       10.0.0.44                0    100      0 300 i
*>i3.2.0.0/24       10.0.0.44                0    100      0 300 i
*> 12.0.0.0         88.88.88.2               0             0 200 i
*>i13.0.0.0/24      10.0.0.44                0    100      0 300 i
*>i99.99.99.0/30    10.0.0.44                0    100      0 300 i
* i172.16.0.0/24    10.0.0.44                1    100      0 i
* i                 10.0.0.33                1    100      0 i
* i                 10.0.0.22                1    100      0 i
```

```
*>                  0.0.0.0                  1         32768 i

Total number of prefixes 7
vyatta@R1:~$
```

R2: show ip bgp summary

Example 1-34 shows the output of the **show ip bgp summary** command for router R2 at this stage of the configuration.

Example 1-34 Verifying route reflector on R2: “show ip bgp summary”

```
vyatta@R2:~$ show ip bgp summary
BGP router identifier 10.0.0.22, local AS number 100
RIB entries 13, using 832 bytes of memory
Peers 1, using 2520 bytes of memory

Neighbor        V   AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.11       4   100      61      55        0    0    0 00:29:21        7

Total number of neighbors 1
vyatta@R2:~$
```

R2: show ip bgp

Example 1-35 shows the output of the **show ip bgp** command for router R2 at this stage of the configuration.

Example 1-35 Verifying route reflector on R2: “show ip bgp”

```
vyatta@R2:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.22
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i3.0.0.0/24       10.0.0.44                0    100      0 300 i
*>i3.1.0.0/24       10.0.0.44                0    100      0 300 i
*>i3.2.0.0/24       10.0.0.44                0    100      0 300 i
*>i12.0.0.0         10.0.0.11                0    100      0 200 i
*>i13.0.0.0/24      10.0.0.44                0    100      0 300 i
```

```
*>i99.99.99.0/30    10.0.0.44                0    100      0 300 i
* i172.16.0.0/24    10.0.0.11                1    100      0 i
*>                  0.0.0.0                  1         32768 i

Total number of prefixes 7
vyatta@R2:~$
```

R3: show ip bgp summary

Example 1-36 shows the output of the **show ip bgp summary** command for router R3 at this stage of the configuration.

Example 1-36 Verifying route reflector on R3: “show ip bgp summary”

```
vyatta@R3:~$ show ip bgp summary
BGP router identifier 10.0.0.33, local AS number 100
RIB entries 13, using 832 bytes of memory
Peers 1, using 2520 bytes of memory

Neighbor        V   AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.11       4   100      60      54        0    0    0 00:32:15        7

Total number of neighbors 1
vyatta@R3:~$
```

R3: show ip bgp

Example 1-37 shows the output of the **show ip bgp** command for router R3 at this stage of the configuration.

Example 1-37 Verifying route reflector on R3: “show ip bgp”

```
vyatta@R3:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.33
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i3.0.0.0/24       10.0.0.44                0    100      0 300 i
*>i3.1.0.0/24       10.0.0.44                0    100      0 300 i
*>i3.2.0.0/24       10.0.0.44                0    100      0 300 i
```

```
*>i12.0.0.0         10.0.0.11                0    100      0 200 i
*>i13.0.0.0/24      10.0.0.44                0    100      0 300 i
*>i99.99.99.0/30    10.0.0.44                0    100      0 300 i
* i172.16.0.0/24    10.0.0.11                1    100      0 i
*>                  0.0.0.0                  1         32768 i

Total number of prefixes 7
vyatta@R3:~$
```

R4: show ip bgp summary

Example 1-38 shows the output of the **show ip bgp summary** command for router R4 at this stage of the configuration.

Example 1-38 Verifying route reflector on R4: “show ip bgp summary”

```
vyatta@R4:~$ show ip bgp summary
BGP router identifier 10.0.0.44, local AS number 100
RIB entries 13, using 832 bytes of memory
Peers 2, using 5040 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.11       4   100      51      54        0    0    0 00:34:25        2
99.99.99.2      4   300      44      48        0    0    0 00:34:19        5

Total number of neighbors 2
vyatta@R4:~$
```

R4: show ip bgp

Example 1-39 shows the output of the **show ip bgp** command for router R4 at this stage of the configuration.

Example 1-39 Verifying route reflector on R4: “show ip bgp”

```
vyatta@R4:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.44
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
```

```
*> 3.0.0.0/24       99.99.99.2               0             0 300 i
*> 3.1.0.0/24       99.99.99.2               0             0 300 i
*> 3.2.0.0/24       99.99.99.2               0             0 300 i
*>i12.0.0.0         10.0.0.11                0    100      0 200 i
*> 13.0.0.0/24      99.99.99.2               0             0 300 i
*> 99.99.99.0/30    99.99.99.2               0             0 300 i
* i172.16.0.0/24    10.0.0.11                1    100      0 i
*>                  0.0.0.0                  1         32768 i

Total number of prefixes 7
vyatta@R4:~$
```

Route Redirection

Route redirection in BGP is performed by means of routing policies. For more information about routing policies, please see the *Vyatta Policy and QoS Reference Guide*.

Chapter 2: Global and Router-Specific Configuration

This chapter describes commands for global and router-specific configuration for BGP.

This chapter presents the following topics:

- Global and Router-Specific Commands

Global and Router-Specific Commands

This chapter contains the following commands.

Configuration Commands		
Global BGP Configuration		
protocols bgp <asn> aggregate-address <ipv4net>	Configuration	Specifies a block of IP addresses to aggregate.
protocols bgp <asn> network <ipv4net>	Configuration	Specifies a network to be advertised by the BGP routing process.
protocols bgp <asn> timers	Configuration	Sets BGP timers globally for the local router.
Router-Specific BGP Configuration		
protocols bgp <asn> parameters always-compare-med	Configuration	Directs the router to compare the MED for paths from neighbors in different autonomous systems.
protocols bgp <asn> parameters bestpath as-path	Configuration	Directs the router to compare the AS paths during best path selection.
protocols bgp <asn> parameters bestpath compare-routerid	Configuration	Directs the router to compare identical routes received from different external peers during best path selection.
protocols bgp <asn> parameters bestpath med	Configuration	Directs the router to compare the Multi Exit Discriminator (MED) among paths learned from confederation peers during best path selection.
protocols bgp <asn> parameters dampening	Configuration	Enables or disables route dampening and sets route dampening values.
protocols bgp <asn> parameters default	Configuration	Sets default routing behaviors for the system.
protocols bgp <asn> parameters deterministic-med	Configuration	Enables or disables enforcing of deterministic MED.
protocols bgp <asn> parameters disable-network-import-check	Configuration	Disables IGP route check for network statements.
protocols bgp <asn> parameters enforce-first-as	Configuration	Enables or disables forcing eBGP peers to list AS number at the beginning of the AS_PATH attribute in incoming updates.
protocols bgp <asn> parameters graceful-restart	Configuration	Enables or disables graceful restart of the BGP process.

protocols bgp <asn> parameters log-neighbor-changes	Configuration	Enables or disables logging of the status of BGP neighbors.
protocols bgp <asn> parameters no-fast-external-failover	Configuration	Enables or disables automatic resetting of BGP sessions for failed links.
protocols bgp <asn> parameters router-id <id>	Configuration	Sets a fixed BGP router ID for the router, overriding the automatic ID selection process.
protocols bgp <asn> parameters scan-time <seconds>	Configuration	Sets the scanning interval for the router.
Operational Commands		
clear ip bgp <address>	Operational	Resets a BGP connection.
clear ip bgp <address> ipv4 unicast	Operational	Resets an IPv4 unicast BGP connection.
clear ip bgp dampening	Operational	Clears BGP route dampening information and unsuppresses suppressed routes.
debug bgp	Operational	Enables or disables debug message generation related to the acquisition of the BGP router ID and the sending and receiving of BGP messages.
debug bgp events	Operational	Enables or disables debug message generation related to BGP events.
debug bgp fsm	Operational	Enables or disables debug message generation related to the BGP Finite State Machine (FSM).
debug bgp keepalives	Operational	Display debugging information related to sending and receiving BGP keep-alive messages.
debug bgp updates	Operational	Display debugging information related to BGP routing updates.
debug bgp zebra	Operational	Display debugging information related to configuration of the Zebra BGP daemon.
show debugging bgp	Operational	Displays BGP protocol debugging flags.
show ip bgp	Operational	Displays BGP routes.
show ip bgp attribute-info	Operational	Displays BGP attribute information.
show ip bgp cidr-only	Operational	Displays BGP routes with CIDR network masks.
show ip bgp community-info	Operational	Displays BGP community information.

show ip bgp community <community>	Operational	Displays BGP routes belonging to the specified BGP community.
show ip bgp community-list <list-name>	Operational	Displays BGP routes permitted by the specified community list.
show ip bgp dampened-paths	Operational	Displays BGP routes that are currently dampened.
show ip bgp filter-list <list-num>	Operational	Displays routes matching a list of autonomous system paths.
show ip bgp flap-statistics	Operational	Displays route flap statistics for BGP routes.
show ip bgp flap-statistics cidr-only	Operational	Displays only route flap statistics for BGP routes with CIDR network masks.
show ip bgp flap-statistics filter-list <list-num>	Operational	Displays route flap statistics for BGP routes matching a list of AS paths.
show ip bgp flap-statistics prefix-list <list-name>	Operational	Displays route flap statistics for BGP routes matching a prefix list.
show ip bgp flap-statistics regexp <expr>	Operational	Displays route flap statistics for BGP routes matching an AS path regular expression.
show ip bgp flap-statistics route-map <map-name>	Operational	Displays route flap statistics for BGP routes matching a route map.
show ip bgp ipv4 unicast	Operational	Displays information for IPv4 unicast BGP routes.
show ip bgp ipv4 unicast cidr-only	Operational	Displays IPv4 unicast BGP routes with CIDR network masks.
show ip bgp ipv4 unicast community <community>	Operational	Displays IPv4 unicast BGP routes belonging to the specified community.
show ip bgp ipv4 unicast community-list <list-name>	Operational	Displays IPv4 unicast BGP routes permitted by the specified community list.
show ip bgp ipv4 unicast paths	Operational	Displays BGP IPv4 unicast path information.
show ip bgp ipv4 unicast prefix-list <list-name>	Operational	Displays IPv4 unicast BGP routes matching a prefix list.
show ip bgp ipv4 unicast regexp <regexp>	Operational	Displays IPv4 unicast BGP routes matching an AS path regular expression.
show ip bgp ipv4 unicast route-map <map-name>	Operational	Displays IPv4 unicast BGP routes matching a route map.
show ip bgp ipv4 unicast statistics	Operational	Displays statistics for BGP IPv4 unicast routes.

show ip bgp memory	Operational	Displays memory usage for BGP.
show ip bgp paths	Operational	Displays all BGP paths.
show ip bgp prefix-list <list-name>	Operational	Displays BGP routes matching a prefix list.
show ip bgp regexp <regexp>	Operational	Displays routes matching an AS path regular expression.
show ip bgp route-map <map-name>	Operational	Displays routes matching a route map.
show ip bgp scan	Operational	Displays BGP scan status.
show ip route bgp	Operational	Displays BGP routes.

clear ip bgp <address>

Resets a BGP connection.

Syntax

clear ip bgp {*ipv4*|*ipv6*} [**in** [**prefix-filter**] | **out** | **rsclient** | **soft** [**in** | **out**]]

Command Mode

Operational mode.

Parameters

ipv4	Resets the connection for the IPv4 BGP peer at the specified address.
ipv6	Resets the connection for the IPv6 BGP peer at the specified address.
in	Optional. Resets the inbound session only.
out	Optional. Resets the outbound session only.
prefix-filter	Optional. Clears the BGP outbound route filter (ORF). This keyword is ignored unless ORF capabilities have been enabled on the local system or received from the sending BGP peer. In this case, a normal inbound soft reset is performed.
rsclient	Optional. Resets only connections in the route server client Routing Information Base (RIB).
soft	Optional. Uses soft reconfiguration for the reset.
in	Optional. Resets with soft reconfiguration only inbound sessions.
out	Optional. Resets with soft reconfiguration only outbound sessions.

Default

Both inbound and outbound sessions are reset.

Usage Guidelines

Use this command to reset a BGP connection.

clear ip bgp <address> ipv4 unicast

Resets an IPv4 unicast BGP connection.

Syntax

clear ip bgp {*ipv4* | *ipv6*} **ipv4 unicast** [**in** [**prefix-filter**] | **out** | **soft** [**in** | **out**]]

Command Mode

Operational mode.

Parameters

ipv4	Resets the connection for the IPv4 BGP peer at the specified address.
ipv6	Resets the connection for the IPv6 BGP peer at the specified address.
in	Optional. Resets inbound sessions only.
out	Optional. Resets outbound sessions only.
prefix-filter	Optional. Clears the BGP outbound route filter (ORF). The **prefix-filter** keyword is ignored unless ORF capabilities have been enabled on the local system or received from the sending BGP peer. In this case, a normal inbound soft reset is performed.
soft	Optional. Uses soft reconfiguration for the reset.
in	Optional. Resets with soft reconfiguration only inbound sessions.
out	Optional. Resets with soft reconfiguration only outbound sessions.

Default

When used without the **soft** option, reset connections are dropped, both inbound and outbound.

.

Usage Guidelines

Use this command to reset an inbound BGP IPv4 unicast session for a given IP address.This forces BGP updates to be generated and new BGP policies to be applied.

Unless the **soft** option is used, all connections are dropped (a “hard reset”): TCP connections are terminated and all routes received from the neighbor are removed from the BGP routing table. Then the connection with the neighbor is reestablished.

If the **soft** option is used, routes from the neighbor are marked as stale but are not immediately removed from the BGP table. Stale routes that are not received from the neighbor when the connection is reestablished are removed from the BGP table at that point.

clear ip bgp dampening

Clears BGP route dampening information and unsuppresses suppressed routes.

Syntax

clear ip bgp dampening [*ipv4* [*ipv4-mask*] | *ipv4net*]

Command Mode

Operational mode.

Parameters

ipv4	Optional. Clears route dampening information for the IPv4 peer at the specified address.
ipv4-mask	Optional. An IPv4 network mask.
ipv4net	Optional. Clears route dampening information for all peers on the specified network. The format is *ip-address/prefix*.

Default

When used with no option, this command clears route dampening information and unsuppresses routes for all BGP peers.

Usage Guidelines

Use this command on a router running BGP to clear information related to route dampening and unsupress routes that are currently suppressed.

debug bgp

Enables or disables debug message generation related to the acquisition of the BGP router ID and the sending and receiving of BGP messages.

Syntax

debug bgp

no debug bgp

undebug bgp

Command Mode

Operational mode.

Parameters

None.

Default

None

Usage Guidelines

Use this command to enable generation of trace-level messages related to the acquisition of the BGP router ID and the sending and receiving of BGP messages.

Use the **no** or **undebug** forms of this command to disable debugging related to the acquisition of the BGP router ID and the sending and receiving of BGP messages.

debug bgp events

Enables or disables debug message generation related to BGP events.

Syntax

debug bgp events

no debug bgp events

undebug bgp events

Command Mode

Operational mode.

Parameters

None.

Default

None

Usage Guidelines

Use this command to enable generation of trace-level messages related to BGP events.

Use the **no** or **undebug** forms of this command to disable debugging of BGP events.

debug bgp fsm

Enables or disables debug message generation related to the BGP Finite State Machine (FSM).

Syntax

debug bgp fsm

no debug bgp fsm

undebug bgp fsm

Command Mode

Operational mode.

Parameters

None.

Default

None

Usage Guidelines

Use this command to enable generation of trace-level messages related to the BGP finite state machine (FSM).

A BGP router uses an FSM consisting of six states, as defined by RFC 1771. The FSM describes how and when the BGP router should make decisions about its operations with other BGP neighbors.

Use the **no** or **undebug** forms of this command to disable debugging of the BGP FSM.

debug bgp keepalives

Display debugging information related to sending and receiving BGP keep-alive messages.

Syntax

debug bgp keepalives

no debug bgp keepalives

undebug bgp keepalives

Command Mode

Operational mode.

Parameters

None.

Default

None

Usage Guidelines

Use this command to enable generation of trace-level messages related to sending and receiving BGP keep-alive messages.

Use the **no** or **undebug** forms of this command to disable debugging of BGP keep-alive messages.

debug bgp updates

Display debugging information related to BGP routing updates.

Syntax

debug bgp updates [**in** | **out**]

no debug bgp updates

undebug bgp updates

Command Mode

Operational mode.

Parameters

in	Optional. Debug information is generated only for inbound routing updates.
out	Optional. Debug information is generated only for outbound routing updates.

Default

Debug messages are generated for both inbound and outbound routing messages.

Usage Guidelines

Use this command to enable generation of trace-level messages related to BGP routing updates.

Use the **no** or **undebug** forms of this command to disable debugging or BGP routing updates.

debug bgp zebra

Display debugging information related to configuration of the Zebra BGP daemon.

Syntax

debug bgp zebra

no debug bgp zebra

undebug bgp zebra

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to enable generation of trace-level messages related to configuration of the Zebra BGP daemon.

Use the **no** or **undebug** forms of this command to disable debugging of the Zebra BGP daemon.

no debug all bgp

Disables all BGP debugging.

Syntax

no debug all bgp

undebug all bgp

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to disable all BGP debug message generation.

protocols bgp <asn>

Creates a BGP instance on the router and locates it within an Autonomous System (AS).

Syntax

set protocols bgp *asn*

delete protocols bgp *asn*

show protocols bgp [*asn*]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. Any peers of this router must be configured to know this AS number—if there is a mismatch, a peering will not be established. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.

Default

None.

Usage Guidelines

Use the **set** form of this command to enable a BGP instance on the router, and set its Autonomous System. All subsequent BGP configuration falls under this configuration node.

Note that you cannot create another BGP instance (that is, you cannot specify a second or further AS) unless you declare multiple BGP instances using the **multiple-instance** command.

Use the **delete** form of this command to disable BGP on the router, removing all BGP configuration.

Use the **show** form of this command to view all BGP configuration.

protocols bgp <asn> aggregate-address <ipv4net>

Specifies a block of IP addresses to aggregate.

Syntax

set protocols bgp *asn* **aggregate-address** *ipv4net* [**as-set** | **summary-only**]

delete protocols bgp *asn* **aggregate-address** *ipv4net*

show protocols bgp *asn* **aggregate-address** [*ipv4net*]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      aggregate-address ipv4net {
         as-set
         summary-only
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides.
ipv4net	Mandatory. The network from which routes are to be aggregated. The format is ip-address/prefix.
as-set	Specifies that the routes resulting from aggregation include the AS set.
summary-only	Specifies that aggregated routes are summarized. These routes will not be announced.

Usage Guidelines

Use the **set** form of this command to specify a contiguous block of IP addresses to aggregate.

Use the **delete** form of this command to delete an aggregate address.

Use the **show** form of this command to view aggregate address configuration settings.

protocols bgp <asn> network <ipv4net>

Specifies a network to be advertised by the BGP routing process.

Syntax

set protocols bgp *asn* **network** *ipv4net* [**backdoor** | **route-map** *map-name*]

delete protocols bgp *asn* **network** *ipv4net* [**backdoor** | **route-map**]

show protocols bgp *asn* **network**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      network ipv4net {
         backdoor
         route-map: text
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
ipv4net	Mandatory. Multi-node. An IPv4 network in the format *ip-address/prefix*. You can advertise to multiple networks by creating multiple **network** configuration nodes.
backdoor	Optional. Indicates that this network is reachable by a backdoor route. A backdoor network is considered to be like a local network, but is not advertised.
route-map *map-name*	Optional. Specifies a configured route map to be used when advertising the network.

Default

None.

Usage Guidelines

Use this command to advertise networks to BGP neighbors.

Use the **set** form of this command to specify a network to be announced via BGP.

Use the **delete** form of this command to remove a network from the list of networks to be announced by BGP.

Use the **show** form of this command to view BGP network advertising configuration settings.

protocols bgp <asn> parameters always-compare-med

Directs the router to compare the MED for paths from neighbors in different autonomous systems.

Syntax

set protocols bgp *asn* **parameters always-compare-med**

delete protocols bgp *asn* **parameters always-compare-med**

show protocols bgp *asn* **parameters**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         always-compare-med
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.

Default

The MED is not compared.

Usage Guidelines

Use the **set** form of this command to have the router compare the Multi Exit Discriminator (MED) for paths from neighbors in different autonomous systems.

The MED is compared only if the AS path for the compared routes is identical.

Use the **delete** form of this command to disable MED comparison.

Use the **show** form of this command to view global BGP configuration settings.

protocols bgp <asn> parameters bestpath as-path

Directs the router to compare the AS paths during best path selection.

Syntax

set protocols bgp *asn* **parameters bestpath as-path** [**confed** | **ignore**]

delete protocols bgp *asn* **parameters bestpath as-path**

show protocols bgp *asn* **parameters bestpath**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         bestpath {
            as-path {
               confed
               ignore
            }
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
confed	Optional. Directs the router to compare the AS paths within a confederation during best path selection.
ignore	Optional. Directs the router to ignore AS_PATH during best path selection.

Default

By default, when making the best-path selection the router does not compare AS_PATHs within a confederation and does not ignore the AS_PATH.

Usage Guidelines

Use the **set** form of this command to direct the router to compare the AS paths during best path selection.

Use the **delete** form of this command to restore the default best-path selection behavior.

Use the **show** form of this command to view BGP best path selection configuration settings.

protocols bgp <asn> parameters bestpath compare-routerid

Directs the router to compare identical routes received from different external peers during best path selection.

Syntax

set protocols bgp *asn* **parameters bestpath compare-routerid**

delete protocols bgp *asn* **parameters bestpath compare-routerid**

show protocols bgp *asn* **parameters bestpath**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         bestpath {
            compare-routerid
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.

Default

By default, when making the best-path selection the router does not compare identical routes received from different external peers.

Usage Guidelines

Use the **set** form of this command to direct the router to compare identical routes received from different external peers during best path selection, and select as the best path the route with the lowest router ID.

Use the **delete** form of this command to restore the default best-path selection behavior.

Use the **show** form of this command to view BGP best path selection configuration settings.

protocols bgp <asn> parameters bestpath med

Directs the router to compare the Multi Exit Discriminator (MED) among paths learned from confederation peers during best path selection.

Syntax

set protocols bgp *asn* **parameters bestpath med** [**confed** | **missing-as-worst**]

delete protocols bgp *asn* **parameters bestpath med** [**confed** | **missing-as-worst**]

show protocols bgp *asn* **parameters bestpath**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         bestpath {
            med {
               confed
               missing-as-worst
            }
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
confed	Optional. Compare the MED among confederation paths
missing-as-worst	Optional. Treat a missing MED as the least preferred one.

Default

By default, when making the best-path selection the router does not consider the MED.

Usage Guidelines

Use the **set** form of this command to direct the router to compare the Multi Exit Discriminator (MED) among paths learned from confederation peers during best-path selection.

Use the **delete** form of this command to restore the default best-path selection behavior.

Use the **show** form of this command to view BGP best path selection configuration settings.

protocols bgp <asn> parameters dampening

Enables or disables route dampening and sets route dampening values.

Syntax

set protocols bgp *asn* **parameters dampening** [**half-life** *minutes* | **re-use** *penalty* | **start-suppress-time** *penalty* | **max-suppress-time** *minutes*]

delete protocols bgp *asn* **parameters dampening** [**half-life** | **re-use** | **start-suppress-time** | **max-suppress-time**]

show protocols bgp *asn* **parameters dampening**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         dampening {
            half-life: 1-45
            max-suppress-time: 1-20000
            re-use: 1-20000
            start-suppress-time: 1-20000
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
half-life *minutes*	Optional. The time period, in minutes, after which the penalty assigned to a route because of flapping is reduced by half. The range is 1 to 45. The default is 15.
max-suppress-time *minutes*	Optional. The maximum time, in minutes, that a route may be suppressed. The range is 1 to 20000. The default is four times the half-life period.

re-use *penalty*	Optional. The reuse threshold. If the penalty for a flapping route is reduced below this point, the route is to be brought back into use. The range is 1 to 20000. The default is 750.
start-suppress-time *penalty*	Optional. The route suppression threshold. If the accumulated penalty for a flapping route reaches this limit, the route is suppressed. The range is 1 to 20000. The default is 2000.

Default

Route dampening is disabled.

Usage Guidelines

Use the **set** form of this command to configure route dampening. When used with no options, this command enables route dampening at the default values.

Use the **delete** form of this command to disable route dampening or reset route dampening parameters to default values.

Use the **show** form of this command to view BGP route dampening configuration settings.

protocols bgp <asn> parameters default

Sets default routing behaviors for the system.

Syntax

set protocols bgp *asn* **parameters default** [**local-pref** *pref* | **no-ipv4-unicast**]

delete protocols bgp *asn* **parameters default** [**local-pref** | **no-ipv4-unicast**]

show protocols bgp *asn* **parameters default**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         default {
            local-pref: u32
            no-ipv4-unicast
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
local-pref *pref*	Optional. Specifies the degree of preference that iBGP peers are to give local routes during BGP best path selection. The higher the value, the more the route is to be preferred. The range is 0 to 4294967295. The default is 100.
no-ipv4-unicast	Optional. Disables the IPv4 unicast address family as the default for peering session establishment. By default, IPv4 address family prefixes are automatically exchanged.

Default

The value of the local-pref attribute is 100. IPv4 unicast is the default address family.

Usage Guidelines

Use the **set** form of this command to override default local route preferences and automatic address family exchanges.

Use the **delete** form of this command to reset the route preferences and address family exchanges to the default.

Use the **show** form of this command to view BGP default routing configuration settings.

protocols bgp <asn> parameters deterministic-med

Enables or disables enforcing of deterministic MED.

Syntax

set protocols bgp *asn* **parameters deterministic-med**

delete protocols bgp *asn* **parameters deterministic-med**

show protocols bgp *asn* **parameters**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         deterministic-med
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.

Default

Deterministic MED is not enforced.

Usage Guidelines

Use the **set** form of this command to enforce the deterministic comparison of the Multi Exit Discriminator (MED) value between all paths received from within the same autonomous system.

Use the **delete** form of this command to disable required MED comparison.

Use the **show** form of this command to view global BGP configuration settings.

protocols bgp <asn> parameters disable-network-import-check

Disables IGP route check for network statements.

Syntax

set protocols bgp *asn* **parameters disable-network-import-check**

delete protocols bgp *asn* **parameters disable-network-import-check**

show protocols bgp *asn* **parameters**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         disable-network-import-check
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.

Default

IGP route check for network statements is enabled.

Usage Guidelines

This command will cause BGP to advertise a network regardless of the Interior Gateway Protocol (IGP) in use.

Use the **set** form of this command to disable IGP route checks for network statements.

Use the **delete** form of this command to enable IGP route checks for network statements.

Use the **show** form of this command to view global BGP configuration settings.

protocols bgp <asn> parameters enforce-first-as

Enables or disables forcing eBGP peers to list AS number at the beginning of the AS_PATH attribute in incoming updates.

Syntax

set protocols bgp *asn* **parameters enforce-first-as**

delete protocols bgp *asn* **parameters enforce-first-as**

show protocols bgp *asn* **parameters**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         enforce-first-as
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.

Default

Enabled.

Usage Guidelines

Use the **set** form of this command to direct the router to enforce listing of an eBGP peer's AS number at the beginning of the AS_PATH.

When this feature is enabled, the router will deny an update received from an external BGP (eBGP) peer unless the AS number is listed at the beginning of the AS_PATH in the incoming update. This prevents “spoof” situations where a misconfigured or unauthorized peer misdirecting traffic by advertising a route as if it were sourced from another autonomous system.

Use the **delete** form of this command to disable this behavior.

Use the **show** form of this command to view global BGP configuration settings.

protocols bgp <asn> parameters graceful-restart

Enables or disables graceful restart of the BGP process.

Syntax

set protocols bgp *asn* **parameters graceful-restart** [**stalepath-time** *seconds*]

delete protocols bgp *asn* **parameters graceful-restart**

show protocols bgp *asn* **parameters**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         graceful-restart {
            stalepath-time: 1-3600
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
stalepath-time *seconds*	Optional. Sets the maximum interval, in seconds, to retain stale paths for a restarting peer. If this interval is exceeded, all stale paths are removed. The range is 1 to 3600. The default is 360. The default value for this attribute is optimal for most deployments, and changing this value can have negative effects on network behavior. We recommend that only experienced network operators change this values.

Default

By default, paths for restarting peers are retained for at most 360 seconds.

Usage Guidelines

Use the **set** form of this command to direct the router to restart gracefully when it is reset.

Use the **delete** form of this command to disable graceful restart.

Use the **show** form of this command to view global BGP configuration settings.

protocols bgp <asn> parameters log-neighbor-changes

Enables or disables logging of the status of BGP neighbors.

Syntax

set protocols bgp *asn* **parameters log-neighbor-changes**

delete protocols bgp *asn* **parameters log-neighbor-changes**

show protocols bgp *asn* **parameters**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         log-neighbor-changes
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.

Default

Disabled.

Usage Guidelines

Use the **set** form of this command to direct the router to log changes in adjacencies for BGP neighbors.

This feature helps detect network problems, by recording when BGP neighbors come up or go down, and when they reset. Log messages are sent to the main log file.

Use the **delete** form of this command to disable logging of neighbor status changes.

Use the **show** form of this command to view global BGP configuration settings.

Even when this feature is disabled, the system tracks neighbor resets; these resets can be seen in the output of the **show ip bgp neighbors** command.

protocols bgp <asn> parameters no-fast-external-failover

Enables or disables automatic resetting of BGP sessions for failed links.

Syntax

set protocols bgp *asn* **parameters no-fast-external-failover**

delete protocols bgp *asn* **parameters no-fast-external-failover**

show protocols bgp *asn* **parameters**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         no-fast-external-failover
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.

Default

Disabled. By default, sessions are automatically reset.

Usage Guidelines

Use the **set** form of this command to disable fast external failover.

When fast external failover is enabled, then BGP sessions for directly adjacent external peers are immediately reset if the link fails.

Use the **delete** form of this command to restore fast external failover.

Use the **show** form of this command to view global BGP configuration settings.

protocols bgp <asn> parameters router-id <id>

Sets a fixed BGP router ID for the router, overriding the automatic ID selection process.

Syntax

set protocols bgp *asn* **parameters router-id** *id*

delete protocols bgp *asn* **parameters router-id** *id*

show protocols bgp *asn* **parameters**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    bgp asn {
        parameters {
            router-id: ipv4
        }
    }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The ID to be used by the router as the BGP router ID.

Default

When router ID is not explicitly set, then the router ID is automatically set to the IP address of the loopback address, if configured. If the loopback address is not configured, the router ID is set to the highest IP address on a physical interface.

Usage Guidelines

Use the **set** form of this command to configure a fixed router ID for the local BGP routing process. This ID will override the router ID automatic set by the system.

Use the **delete** form of this command to remove the fixed router ID and restore the automatically selected ID.

Use the **show** form of this command to view global BGP configuration settings.

protocols bgp <asn> parameters scan-time <seconds>

Sets the scanning interval for the router.

Syntax

set protocols bgp *asn* **parameters scan-time** *seconds*

delete protocols bgp *asn* **parameters scan-time** *seconds*

show protocols bgp *asn* **parameters**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         scan-time: u32
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
seconds	Mandatory. The interval, in seconds, at which the router scans for BGP routing information. The range is 5 to 60. The default is 15.

Default

Scans take place every 15 seconds.

Usage Guidelines

Use the **set** form of this command to set the interval at which the router scans for BGP routing information.

Use the **delete** form of this command to reset the scanning interval to the default.

Use the **show** form of this command to view global BGP configuration settings.

protocols bgp <asn> timers

Sets BGP timers globally for the local router.

Syntax

set protocols bgp *asn* **timers** [**keepalive** *seconds* | **holdtime** *seconds*]

delete protocols bgp *asn* **timers** [**keepalive** | **holdtime**]

show protocols bgp *asn* **timers**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      timers {
         keepalive: 1-65535
         holdtime: 0, 4-65535
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
keepalive *seconds*	Optional. The frequency, in seconds, with which the local router sends keep-alive messages to neighbors. The range is 1 to 65535. The default is 60.
holdtime *seconds*	Optional. The maximum interval, in seconds, after which if the local router has not received a keep-alive message from the neighbor, a neighbor is declared dead. The range is 0 and 4 to 65535, where 0 disables the holdtime timer. The default is 180.

Default

The default for the keep-alive timer is 60 seconds. The default for the holdtime timer is 180 seconds.

Usage Guidelines

Use the **set** form of this command to set global BGP timers for monitoring the health of remote peers. These timers will be applied to all remote peers unless a neighbor has timers explicitly configured for it. Timers explicitly specified for a neighbor override the timers set globally.

Use the **delete** form of this command to restore global BGP timers to default values.

Use the **show** form of this command to view global BGP timer configuration settings.

show debugging bgp

Displays BGP protocol debugging flags.

Syntax

show debugging bgp

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display BGP protocol debugging flags.

show ip bgp

Displays BGP routes.

Syntax

show ip bgp [*ipv4* | *ipv4net* [**longer-prefixes**] | **summary**]

Command Mode

Operational mode.

Parameters

ipv4	Optional. Displays routes for the neighbor at the specified IPv4 address.
ipv4net	Optional. Displays routes for the specified IPv4 network.
longer-prefixes	Optional. Displays any routes more specific than the one specified.
summary	Optional. Shows summary BGP route information.

Default

Displays all BGP routes by default.

Usage Guidelines

Use this command to display the BGP routing table.

show ip bgp attribute-info

Displays BGP attribute information.

Syntax

show ip bgp attribute-info

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display BGP attribute information.

show ip bgp cidr-only

Displays BGP routes with CIDR network masks.

Syntax

show ip bgp cidr-only

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display only routes with non-natural network masks; that is, Classless Inter Domain Routing network masks.

show ip bgp community-info

Displays BGP community information.

Syntax

show ip bgp community-info

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display BGP community information.

show ip bgp community <community>

Displays BGP routes belonging to the specified BGP community.

Syntax

show ip bgp community *community* [**exact-match**]

Command Mode

Operational mode.

Parameters

community	Mandatory. A BGP community identifier in the form AA:NN (where AA and NN are in the range of 0-65535), one of the well-known BGP communities **local-AS**, **no-export**, or **no-advertise**, or a space-separated list of up to four community identifiers.
exact-match	Optional. Displays only routes that have an exact match.

Default

None.

Usage Guidelines

Use this command to display the BGP routes belonging to up to four BGP communities.

show ip bgp community-list <list-name>

Displays BGP routes permitted by the specified community list.

Syntax

show ip bgp community-list *list-name* [**exact-match**]

Command Mode

Operational mode.

Parameters

list-name	Mandatory. A preconfigured list of BGP communities.
exact-match	Optional. Displays only route that have an exact match.

Default

None.

Usage Guidelines

Use this command to display the BGP routes permitted by the specified community list.

show ip bgp dampened-paths

Displays BGP routes that are currently dampened.

Syntax

show ip bgp dampened-paths

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display the BGP routes that are currently dampened.

show ip bgp filter-list <list-num>

Displays routes matching a list of autonomous system paths.

Syntax

show ip bgp filter-list *list-num*

Command Mode

Operational mode.

Parameters

list-num	Mandatory. The number of a preconfigured autonomous system path access list. The range is 1 to 500.

Default

None.

Usage Guidelines

Use this command to filter displayed routes according to preconfigured access list of autonomous system paths.

BGP filter lists are defined using the **policy** command. For information about creating filter lists, see the *Vyatta Policy and QoS Reference Guide*.

show ip bgp flap-statistics

Displays route flap statistics for BGP routes.

Syntax

show ip bgp flap-statistics [*ipv4* | *ipv4net* [**longer-prefixes**]

Command Mode

Operational mode.

Parameters

ipv4	Optional. Displays flap statistics for routes matching the specified IPv4 address.
ipv4net	Optional. Displays flap statistics for routes matching the specified IPv4 network.
longer-prefixes	Optional. Displays any routes more specific than the one specified.

Default

Displays route flap statistics for all BGP routes.

Usage Guidelines

Use this command to display statistics for flapping BGP routes.

show ip bgp flap-statistics cidr-only

Displays only route flap statistics for BGP routes with CIDR network masks.

Syntax

show ip bgp flap-statistics cidr-only

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display route flap statistics for BGP routes with non-natural network masks; that is, Classless Inter Domain Routing (CIDR) network masks.

show ip bgp flap-statistics filter-list <list-num>

Displays route flap statistics for BGP routes matching a list of AS paths.

Syntax

show ip bgp flap-statistics filter-list *list-num*

Command Mode

Operational mode.

Parameters

list-num	Mandatory. The number of a defined list of AS paths. The range is 1 to 500.

Default

None.

Usage Guidelines

Use this command to display route flap statistics for BGP routes matching a preconfigured access list of autonomous system (AS) paths.

AS path filter lists are configured using the **policy** command. For information about creating path filter lists, see the *Vyatta Policy and QoS Reference Guide*.

show ip bgp flap-statistics prefix-list <list-name>

Displays route flap statistics for BGP routes matching a prefix list.

Syntax

show ip bgp flap-statistics prefix-list *list-name*

Command Mode

Operational mode.

Parameters

list-name	Mandatory. The name of a defined prefix list.

Default

None.

Usage Guidelines

Use this command to display BGP statistics for BGP routes matching a list of network prefixes.

Prefix lists are configured using using the **policy** command. For information about creating filter lists, see the *Vyatta Policy and QoS Reference Guide*.

show ip bgp flap-statistics regexp <expr>

Displays route flap statistics for BGP routes matching an AS path regular expression.

Syntax

show ip bgp flap-statistics regexp *expr*

Command Mode

Operational mode.

Parameters

expr	Mandatory. A POSIX-style regular expression representing a set of AS paths.

Default

Displays flap statistics for all BGP routes by default.

Usage Guidelines

Use this command to display route flap statistics for BGP routes matching a regular expression representing a set of autonomous system (AS) paths.

show ip bgp flap-statistics route-map <map-name>

Displays route flap statistics for BGP routes matching a route map.

Syntax

show ip bgp flap-statistics route-map *map-name*

Command Mode

Operational mode.

Parameters

map-name	Optional. The name of a defined route map.

Default

Displays flap statistics for all BGP routes by default.

Usage Guidelines

Use this command to display route flap statistics for BGP routes matching a preconfigured route map.

Route maps are configured using the **policy** command. For information about creating route maps, see the *Vyatta Policy and QoS Reference Guide*.

show ip bgp ipv4 unicast

Displays information for IPv4 unicast BGP routes.

Syntax

show ip bgp ipv4 unicast [*ipv4* | *ipv4net* [**longer-prefixes**] | **summary**]

Command Mode

Operational

Parameters

ipv4	Optional. Show BGP information for the specified address.
ipv4net	Optional. Show BGP information for the specified network.
longer-prefixes	Optional. Displays any routes more specific than the one specified.
summary	Optional. Displays summary IPv4 unicast route information.

Default

Displays all IPv4 unicast BGP routes by default.

Usage Guidelines

Use this command to display IPv4 unicast routes in the BGP routing table.

show ip bgp ipv4 unicast cidr-only

Displays IPv4 unicast BGP routes with CIDR network masks.

Syntax

show ip bgp ipv4 unicast cidr-only

Command Mode

Operational

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display only BGP IPv4 unicast routes with non-natural network masks; that is, Classless Inter Domain Routing (CIDR) network masks.

show ip bgp ipv4 unicast community <community>

Displays IPv4 unicast BGP routes belonging to the specified community.

Syntax

show ip bgp ipv4 unicast community *community* [**exact-match**]

Command Mode

Operational mode.

Parameters

community	Mandatory. A BGP community identifier in the form AA:NN (where AA and NN are in the range of 0-65535), one of the well-known BGP communities **local-AS**, **no-export**, or **no-advertise**, or a space-separated list of up to four community identifiers.
exact-match	Optional. Displays only routes that have an exact match.

Default

None.

Usage Guidelines

Use this command to display BGP IPv4 unicast routes belonging to up to four BGP communities.

show ip bgp ipv4 unicast community-list <list-name>

Displays IPv4 unicast BGP routes permitted by the specified community list.

Syntax

show ip bgp ipv4 unicast community-list *list-name* [**exact-match**]

Command Mode

Operational mode.

Parameters

list-name	Mandatory. A preconfigured list of BGP communities.
exact-match	Optional. Displays routes that have an exact match.

Default

None.

Usage Guidelines

Use this command to display BGP IPv4 unicast routes permitted by the specified community list.

show ip bgp ipv4 unicast paths

Displays BGP IPv4 unicast path information.

Syntax

show ip bgp ipv4 unicast paths

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display BGP IPv4 unicast path information.

show ip bgp ipv4 unicast prefix-list <list-name>

Displays IPv4 unicast BGP routes matching a prefix list.

Syntax

show ip bgp ipv4 unicast prefix-list *list-name*

Command Mode

Operational mode.

Parameters

list-name	Mandatory. Name of a defined prefix list.

Default

None.

Usage Guidelines

Use this command to display BGP IPv4 unicast routes matching a preconfigured prefix list.

Prefix lists are configured using the **policy** command. For information about creating prefix lists, see the *Vyatta Policy and QoS Reference Guide*.

show ip bgp ipv4 unicast regexp <regexp>

Displays IPv4 unicast BGP routes matching an AS path regular expression.

Syntax

show ip bgp ipv4 unicast regexp *regexp*

Command Mode

Operational mode.

Parameters

regexp	Mandatory. A POSIX-style regular expression representing a set of AS paths.

Default

None.

Usage Guidelines

Use this command to display BGP IPv4 unicast routes matching the specified AS path regular expression.

show ip bgp ipv4 unicast route-map <map-name>

Displays IPv4 unicast BGP routes matching a route map.

Syntax

show ip bgp ipv4 unicast route-map *map-name*

Command Mode

Operational mode.

Parameters

map-name	Mandatory. Name of a defined route map.

Default

None.

Usage Guidelines

Use this command to display IPv4 unicast BGP routes matching a preconfigured route map.

Route maps are defined using the **policy** command. For information about creating route maps, see the *Vyatta Policy and QoS Reference Guide*.

show ip bgp ipv4 unicast statistics

Displays statistics for BGP IPv4 unicast routes.

Syntax

show ip bgp ipv4 unicast statistics

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display BGP IPv4 unicast statistics.

show ip bgp memory

Displays memory usage for BGP.

Syntax

show ip bgp memory

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display the amount of memory being used for BGP, including the RIB, cache entries, attributes, AS-PATH entries, and hashes.

show ip bgp paths

Displays all BGP paths.

Syntax

show ip bgp paths

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display all BGP paths.

show ip bgp prefix-list <list-name>

Displays BGP routes matching a prefix list.

Syntax

show ip bgp prefix-list *list-name*

Command Mode

Operational mode.

Parameters

list-name	Mandatory. Name of a defined prefix list.

Default

None.

Usage Guidelines

Use this command to display routes that match a preconfigured prefix list.

Prefix lists are configured using the **policy** command. For information about creating prefix lists, see the *Vyatta Policy and QoS Reference Guide*.

show ip bgp regexp <regexp>

Displays routes matching an AS path regular expression.

Syntax

show ip bgp regexp *regexp*

Command Mode

Operational mode.

Parameters

regexp	Mandatory. A POSIX-style regular expression representing a set of AS paths.

Default

None.

Usage Guidelines

Use this command to display routes matching a regular expression representing an autonomous system (AS) path list.

show ip bgp route-map <map-name>

Displays routes matching a route map.

Syntax

show ip bgp route-map *map-name*

Command Mode

Operational mode.

Parameters

map-name	Optional. The name of a defined route map.

Default

None.

Usage Guidelines

Use this command to display routes matching a preconfigured route map.

Route maps are defined using the **policy** command. For information about creating route maps, see the *Vyatta Policy and QoS Reference Guide*.

show ip bgp scan

Displays BGP scan status.

Syntax

show ip bgp scan

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display BGP scan status.

show ip route bgp

Displays BGP routes.

Syntax

show ip route bgp

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display BGP routes.

Examples

Example 2-1 shows BGP routes.

Example 2-1 "show ip route bgp": Displaying BGP routes

```
root@vyatta> show ip route bgp
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

B   10.1.0.0/24 [20/75] via 10.1.0.4 inactive, 1d00h46m
B>* 10.100.100.4/32 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h46m
B>* 10.104.104.4/32 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h46m
B>* 172.16.0.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.20.0.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.20.1.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.20.2.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.20.3.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.20.4.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.20.5.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.20.6.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.20.7.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
```

```
B>* 172.20.8.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.20.9.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.21.0.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.22.0.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.23.0.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.24.0.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.25.0.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.26.0.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.27.0.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.28.0.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 172.29.0.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 1d00h36m
B>* 192.168.3.0/24 [20/75] via 10.1.0.4 (recursive via 10.3.0.1), 05:15:56
root@vyatta>
```

Chapter 3: Route Reflection

This chapter describes commands for for BGP route reflection.

This chapter presents the following topics:

- Route Reflection Commands

Route Reflection Commands

This chapter contains the following commands.

Configuration Commands		
protocols bgp <asn> neighbor <id> route-reflector-client	Configuration	Defines the local router as a BGP route reflector, with the neighbor as a route reflector client.
protocols bgp <asn> parameters cluster-id <id>	Configuration	Sets the cluster ID for a BGP route reflection cluster.
protocols bgp <asn> parameters no-client-to-client-reflection	Configuration	Enables or disables route reflection from a BGP route reflector to clients.
Operational Commands		
None.		

protocols bgp <asn> neighbor <id> route-reflector-client

Defines the local router as a BGP route reflector, with the neighbor as a route reflector client.

Syntax

set protocols bgp *asn* **neighbor** *id* **route-reflector-client**

delete protocols bgp *asn* **neighbor** *id* **route-reflector-client**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         route-reflector-client
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory.The IP address of a BGP neighbor or the name of a BGP peer group.

Default

There is no route reflector in the autonomous system.

Usage Guidelines

Use the **set** form of this command to make the local router a BGP route reflector, and to designate the specified neighbor as a route reflector client.

Another technology designed to help ASs with large numbers of iBGP peers is route reflection. In a standard BGP implementation, all iBGP peers must be fully meshed. because of this requirement, when an iBGP peer learns a route from another iBGP peer, the receiving router does not forward the route to any of its iBGP peers, since these routers should have learned the route directly from the announcing router.

In a route reflector environment the iBGP peers are no longer fully meshed. Instead, each iBGP peer has an iBGP connection to one or more route reflectors (RRs). Routers configured with a connection to an RR server are referred to as RR clients. Only the RR server is configured to be aware that the RR client is part of an RR configuration; from the RR client's point of view, it is configured normally, and does not have any awareness that it is part of a RR configuration.

In route reflection, internal peers of an RR server are categorized into two types:

- **Client peers.** The RR server and its client peers form a cluster. Within a cluster, client peers need not be fully meshed, but must have an iBGP connection to at least one RR in the cluster.
- **Non-client peers.** Non-client peers, including the RR server, must be fully meshed.

An RR environment is unlike a regular environment, where iBGP peers never forward a route update to other iBGP peers (which is the reason why each iBGP peer must peer with all other peers). When an RR server receives an iBGP update from an RR client, these route updates can also be sent to all other RR clients. When an RR server receives a route update from a peer, it selects the best path based on its path selection rule. After the best path is selected, the RR server chooses its action depending on the type of the peer from which it learned the best path.

- If the route was learned from a client peer, the RR reflects the route to both client and non-client peers. All iBGP updates from client peers are reflected to all other client peers in the cluster. This is done regardless of whether the update was the best path for the RR itself.
- If the route was learned from a non-client iBGP peer, it is reflected out to all RR client peers.
- If the route was learned from an eBGP peer, the route is reflected to all RR clients and all non-clients.

Use the **delete** form of this command to remove the neighbor as a route reflector client.

Use the **show** form of this command to view BGP neighbor configuration settings.

Usage Guidelines

Use the **set** form of this command to disable route reflection from a BGP route reflector to clients. When client-to-client route reflection is enabled, the configured route reflector reflects routes from one client to other clients.

Use the **delete** form of this command to enable client-to-client route reflection.

Use the **show** form of this command to view global BGP configuration settings.

protocols bgp <asn> parameters cluster-id <id>

Sets the cluster ID for a BGP route reflection cluster.

Syntax

set protocols bgp *asn* **parameters cluster-id** *id*

delete protocols bgp *asn* **parameters cluster-id** *id*

show protocols bgp *asn* **parameters**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         cluster-id: ipv4
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. A network address uniquely identifying the route reflection cluster.

Default

None.

Usage Guidelines

Use the **set** form of this command to specify the route reflection cluster ID for an internal route reflection cluster.

When there is only one route reflector in a cluster, the cluster uses the router ID of the route reflector as an ID. If more than one route reflector is deployed in a cluster (for example, to provide redundancy), you must provide an ID for the cluster. In this case, the cluster ID is assigned to every route reflector in the cluster.

Use the **delete** form of this command to remove a cluster ID.

Use the **show** form of this command to view global BGP configuration settings.

protocols bgp <asn> parameters no-client-to-client-reflection

Enables or disables route reflection from a BGP route reflector to clients.

Syntax

set protocols bgp *asn* **parameters no-client-to-client-reflection**

delete protocols bgp *asn* **parameters no-client-to-client-reflection**

show protocols bgp *asn* **parameters**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         no-client-to-client-reflection
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.

Default

By default client-to-client-reflection is enabled.

Usage Guidelines

Use the **set** form of this command to disable route reflection from a BGP route reflector to clients. When client-to-client route reflection is enabled, the configured route reflector reflects routes from one client to other clients.

Use the **delete** form of this command to enable client-to-client route reflection.

Use the **show** form of this command to view global BGP configuration settings.

Chapter 4: Confederations

This chapter describes commands for BGP confederations.

This chapter presents the following topics:

- Confederation Commands

Confederation Commands

This chapter contains the following commands.

Configuration Commands		
protocols bgp <asn> parameters confederation identifier <asn>	Configuration	Defines a BGP confederation.
protocols bgp <asn> parameters confederation peers <asn>	Configuration	Defines the autonomous systems that make up a BGP confederation.
Operational Commands		
None.		

protocols bgp <asn> parameters confederation identifier <asn>

Defines a BGP confederation.

Syntax

set protocols bgp *asn* **parameters confederation identifier** *asn*

delete protocols bgp *asn* **parameters confederation identifier** *asn*

show protocols bgp *asn* **parameters confederation**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         confederation {
            identifier: u32
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
identifier *asn*	Mandatory. The AS number of the BGP confederation. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.

Default

None.

Usage Guidelines

Use the **set** form of this command to create a BGP confederation.

Use the **delete** form of this command to remove the BGP confederation.

Use the **show** form of this command to view BGP confederation configuration settings.

protocols bgp <asn> parameters confederation peers <asn>

Defines the autonomous systems that make up a BGP confederation.

Syntax

set protocols bgp *asn* **parameters confederation peers** *asn* [*asn... asn*]

delete protocols bgp *asn* **parameters confederation peers** *asn* [*asn... asn*]

show protocols bgp *asn* **parameters confederation**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      parameters {
         confederation {
            peers: u32
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides.
peers *asn*	Mandatory. The subautonomous systems that will make up the BGP confederation. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems. Multiple ASs can be specified in a space-separated list.

Default

None.

Usage Guidelines

Use the **set** form of this command to list the subautonomous systems that will be the members of a BGP confederation. To a peer outside the confederation, the confederation appears as a single autonomous system.

Use the **delete** form of this command to remove an AS from a confederation.

Use the **show** form of this command to view BGP confederation configuration settings.

Chapter 5: Neighbors and Peer Groups

This chapter describes commands for BGP neighbors and peer groups.

This chapter presents the following topics:

- Neighbor and Peer Group Commands

Neighbor and Peer Group Commands

This chapter contains the following commands.

Configuration Commands		
protocols bgp <asn> neighbor <id>	Configuration	Defines a BGP neighbor.
protocols bgp <asn> neighbor <id> advertisement-interval <seconds>	Configuration	Sets the minimum interval for BGP route updates.
protocols bgp <asn> neighbor <id> allowas-in	Configuration	Allows or disallows receiving BGP advertisements containing the AS path of the local router.
protocols bgp <asn> neighbor <id> attribute-unchanged	Configuration	Allows the router to send updates to a neighbor with unchanged attributes.
protocols bgp <asn> neighbor <id> capability dynamic	Configuration	Advertises support for dynamic update of BGP capabilities advertised and received from this neighbor.
protocols bgp <asn> neighbor <id> capability orf	Configuration	Enables or disables forwarding of the default route to a BGP neighbor.
protocols bgp <asn> neighbor <id> description <desc>	Configuration	Provides a brief description for a BGP neighbor.
protocols bgp <asn> neighbor <id> disable-send-community	Configuration	Disables sending of community attributes to the specified neighbor.
protocols bgp <asn> neighbor <id> distribute-list export <acl-num>	Configuration	Applies an access list to filter outbound routing updates to this neighbor.
protocols bgp <asn> neighbor <id> distribute-list import <acl-num>	Configuration	Applies an access list to filter inbound routing updates from this neighbor.
protocols bgp <asn> neighbor <id> ebgp-multihop <ttl>	Configuration	Allows eBGP neighbors not on directly connected networks.
protocols bgp <asn> neighbor <id> filter-list export <acl-num>	Configuration	Applies a filter list to routing updates to this neighbor.
protocols bgp <asn> neighbor <id> filter-list import <acl-num>	Configuration	Applies a filter list to routing updates from this neighbor.
protocols bgp <asn> neighbor <id> local-as	Configuration	Defines a local autonomous system number for eBGP peerings.
protocols bgp <asn> neighbor <id> maximum-prefix <max-num>	Configuration	Sets the maximum number of prefixes to accept from this neighbor before that neighbor is taken down.

protocols bgp <asn> neighbor <id> nexthop-self	Configuration	Sets the local router as the next hop for this neighbor.
protocols bgp <asn> neighbor <id> override-capability	Configuration	Overrides capability negotiation to allow a peering session to be established with a neighbor that does not support capability negotiation.
protocols bgp <asn> neighbor <id> passive	Configuration	Directs the router not to initiate connections with this neighbor.
protocols bgp <asn> neighbor <id> password <pwd>	Configuration	Specifies the BGP MD5 password.
protocols bgp <asn> neighbor <id> peer-group	Configuration	Creates a peer group, or assigns a neighbor as a member of the specified peer group.
protocols bgp <asn> neighbor <id> port <port-num>	Configuration	Specifies the port on which the neighbor is listening for BGP signals.
protocols bgp <asn> neighbor <id> prefix-list export <list-name>	Configuration	Applies an prefix list to filter updates to this neighbor.
protocols bgp <asn> neighbor <id> prefix-list import <list-name>	Configuration	Applies an prefix list to filter updates from this neighbor.
protocols bgp <asn> neighbor <id> remote-as <asn>	Configuration	Specifies the autonomous system number of the neighbor.
protocols bgp <asn> neighbor <id> remove-private-as	Configuration	Directs the router to remove private AS numbers from updates sent to this neighbor.
protocols bgp <asn> neighbor <id> route-map export <map-name>	Configuration	Applies a route map to filter updates to this neighbor.
protocols bgp <asn> neighbor <id> route-map import <map-name>	Configuration	Applies a route map to filter updates to or from this neighbor.
protocols bgp <asn> neighbor <id> shutdown	Configuration	Administratively shuts down a BGP neighbor.
protocols bgp <asn> neighbor <id> soft-reconfiguration	Configuration	Directs the router to store received routing updates.
protocols bgp <asn> neighbor <id> strict-capability-match	Configuration	Directs the router to strictly match the capabilities of the neighbor.
protocols bgp <asn> neighbor <id> timers	Configuration	Sets BGP timers for this neighbor.
protocols bgp <asn> neighbor <id> unsuppress-map <map-name>	Configuration	Directs the router to selectively advertise routes suppressed by aggregating addresses, based on a route map.

protocols bgp <asn> neighbor <id> update-source <addr>	Configuration	Specifies the source IP address for routing updates.
protocols bgp <asn> neighbor <id> weight <weight>	Configuration	Defines a default weight for routes from this neighbor.
Operational Commands		
clear ip bgp external ipv4 unicast	Operational	Resets peering sessions for eBGP neighbors.
clear ip bgp peer-group <group-name>	Operational	Resets IPv4 unicast peering sessions for eBGP neighbors.
clear ip bgp peer-group <group-name> ipv4 unicast	Operational	Resets sessions for all members of a peer group.
BGP Show Commands		
show ip bgp ipv4 unicast neighbors	Operational	Displays information for IPv4 unicast BGP neighbors.
show ip bgp ipv4 unicast neighbors <ipv4> advertised-routes	Operational	Displays advertised BGP IPv4 unicast routes for a BGP neighbor.
show ip bgp ipv4 unicast neighbors <ipv4> prefix-counts	Operational	Displays IPv4 unicast prefix-counts for a BGP neighbor.
show ip bgp ipv4 unicast neighbors <ipv4> received prefix-filter	Operational	Displays the IPv4 unicast prefix-lists received from a BGP neighbor.
show ip bgp ipv4 unicast neighbors <ipv4> received-routes	Operational	Displays the IPv4 unicast routes received from a BGP neighbor.
show ip bgp ipv4 unicast neighbors <ipv4> routes	Operational	Displays IPv4 unicast received and accepted routes from a BGP neighbor.
show ip bgp neighbors	Operational	Displays BGP neighbor information.
show ip bgp neighbors <ipv4> advertised-routes	Operational	Displays advertised routes for a BGP neighbor.
show ip bgp neighbors <ipv4> dampened-routes	Operational	Displays dampened routes to a BGP neighbor.
show ip bgp neighbors <ipv4> flap-statistics	Operational	Displays route flap statistics for routes learned from a BGP neighbor.
show ip bgp neighbors <ipv4> prefix-counts	Operational	Displays prefix counts for a BGP neighbor.
show ip bgp neighbors <ipv4> received prefix-filter	Operational	Displays prefix lists received from a BGP neighbor.
show ip bgp neighbors <ipv4> received-routes	Operational	Displays routes received from a BGP neighbor.

show ip bgp neighbors <ipv4> routes	**Operational**	**Displays all received and accepted routes from a BGP neighbor.**

Some command for working with BGP neighbors are described in other chapters.

Related Commands Documented Elsewhere	
protocols bgp <asn> parameters log-neighbor-changes	See "Chapter 2: Global and Router-Specific Configuration."
protocols bgp <asn> neighbor <id> route-reflector-client	See "Chapter 3: Route Reflection."
show ip bgp view <view-name> neighbors	See "Chapter 7: Route Server."
protocols bgp <asn> parameters confederation peers <asn>	See "Chapter 4: Confederations."

clear ip bgp external

Resets peering sessions for eBGP neighbors.

Syntax

clear ip bgp external [**in** [**prefix-filter**] | **out** | **soft** [**in** | **out**]]

Command Mode

Operational mode.

Parameters

in	Optional. Resets inbound sessions only.
out	Optional. Resets outbound sessions only.
prefix-filter	Optional. Clears the BGP outbound route filter (ORF). The **prefix-filter** keyword is ignored unless ORF capabilities have been enabled on the local system or received from the sending BGP peer. In this case, a normal inbound soft reset is performed.
soft	Optional. Uses soft reconfiguration for the reset.
in	Optional. Resets with soft reconfiguration only inbound sessions.
out	Optional. Resets with soft reconfiguration only outbound sessions.

Default

When used without the **soft** option, reset connections are dropped, both inbound and outbound.

Usage Guidelines

Use this command on a router running BGP to reset sessions for external BGP (eBGP) neighbors. This forces BGP updates to be generated and new BGP policies to be applied.

Unless the **soft** option is used, all connections are dropped (a “hard reset”): TCP connections are terminated and all routes received from the neighbor are removed from the BGP routing table. Then the connection with the neighbor is reestablished.

If the **soft** option is used, routes from the neighbor are marked as stale but are not immediately removed from the BGP table. Stale routes that are not received from the neighbor when the connection is reestablished are removed from the BGP table at that point.

clear ip bgp external ipv4 unicast

Resets IPv4 unicast peering sessions for eBGP neighbors.

Syntax

clear ip bgp external ipv4 unicast [in [**prefix-filter**] | **out** | **soft** [**in** | **out**]

Command Mode

Operational mode.

Parameters

in	Optional. Resets inbound sessions only.
out	Optional. Resets outbound sessions only.
prefix-filter	Optional. Clears the BGP outbound route filter (ORF). The **prefix-filter** keyword is ignored unless ORF capabilities have been enabled on the local system or received from the sending BGP peer. In this case, a normal inbound soft reset is performed.
soft	Optional. Uses soft reconfiguration for the reset.
in	Optional. Resets with soft reconfiguration only inbound sessions.
out	Optional. Resets with soft reconfiguration only outbound sessions.

Default

When used without the **soft** option, reset connections are dropped, both inbound and outbound.

Usage Guidelines

Use this command to reset inbound IPv4 unicast peering sessions for eBGP neighbors. This forces BGP updates to be generated and new BGP policies to be applied.

Unless the **soft** option is used, all connections are dropped (a “hard reset”): TCP connections are terminated and all routes received from the neighbor are removed from the BGP routing table. Then the connection with the neighbor is reestablished.

If the **soft** option is used, routes from the neighbor are marked as stale but are not immediately removed from the BGP table. Stale routes that are not received from the neighbor when the connection is reestablished are removed from the BGP table at that point.

clear ip bgp peer-group <group-name>

Resets sessions for all members of a peer group.

Syntax

clear ip bgp peer-group *group-name* [**in** [**prefix-filter**] | **out** | **soft** [**in** | **out**]]

Command Mode

Operational mode.

Parameters

group-name	Mandatory. The name of a defined BGP peer group.
in	Optional. Resets inbound sessions only.
out	Optional. Resets outbound sessions only.
prefix-filter	Optional. Clears the BGP outbound route filter (ORF). This keyword is ignored unless ORF capabilities have been enabled on the local system or received from the sending BGP peer. In this case, a normal inbound soft reset is performed.
soft	Optional. Uses soft reconfiguration for the reset.
in	Optional. Resets with soft reconfiguration only inbound sessions.
out	Optional. Resets with soft reconfiguration only outbound sessions.

Default

When used without the **soft** option, reset connections are dropped, both inbound and outbound.

Usage Guidelines

Use this command to reset sessions for all members of a peer group. This forces BGP updates to be generated and new BGP policies to be applied.

Unless the **soft** option is used, all connections are dropped (a “hard reset”): TCP connections are terminated and all routes received from the neighbor are removed from the BGP routing table. Then the connection with the neighbor is reestablished.

If the **soft** option is used, routes from the neighbor are marked as stale but are not immediately removed from the BGP table. Stale routes that are not received from the neighbor when the connection is reestablished are removed from the BGP table at that point.

clear ip bgp peer-group <group-name> ipv4 unicast

Resets IPv4 unicast sessions for all members of a peer group.

Syntax

clear ip bgp peer-group *group-name* **ipv4 unicast [in** [**prefix-filter**] | **out** | **soft** [**in** | **out**]]

Command Mode

Operational mode.

Parameters

group-name	Mandatory. The name of a defined BGP peer group.
in	Optional. Resets inbound sessions only.
out	Optional. Resets outbound sessions only.
prefix-filter	Optional. Clears the BGP outbound route filter (ORF). This keyword is ignored unless ORF capabilities have been enabled on the local system or received from the sending BGP peer. In this case, a normal inbound soft reset is performed.
soft	Optional. Uses soft reconfiguration for the reset.
in	Optional. Resets with soft reconfiguration only inbound sessions.
out	Optional. Resets with soft reconfiguration only outbound sessions.

Default

None.

Usage Guidelines

Use this command to reset inbound unicast sessions for all members of a peer group. This forces BGP updates to be generated and new BGP policies to be applied.

Unless the **soft** option is used, all connections are dropped (a “hard reset”): TCP connections are terminated and all routes received from the neighbor are removed from the BGP routing table. Then the connection with the neighbor is reestablished.

If the **soft** option is used, routes from the neighbor are marked as stale but are not immediately removed from the BGP table. Stale routes that are not received from the neighbor when the connection is reestablished are removed from the BGP table at that point.

protocols bgp <asn> neighbor <id>

Defines a BGP neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id*

delete protocols bgp *asn* **neighbor** *id*

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {}
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. Multi-node. The IP address of a BGP neighbor. You can define multiple BGP neighbors by creating multiple **neighbor** configuration nodes.

Default

None.

Usage Guidelines

Use the **set** form of this command to define a BGP neighbor.

Use the **delete** form of this command to remove a BGP neighbor.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> advertisement-interval <seconds>

Sets the minimum interval for BGP route updates.

Syntax

set protocols bgp *asn* **neighbor** *id* **advertisement-interval** *seconds*

delete protocols bgp *asn* **neighbor** *id* **advertisement-interval**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         advertisement-interval: u32
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
seconds	Mandatory. The minimum interval, in seconds, between BGP routing updates to this neighbor. The range is 0 to 600. The default is 30 for eBGP peers and 5 for iBGP peers.

Default

The default advertisement interval is 30 seconds for eBGP peers and 5 seconds for iBGP peers.

Usage Guidelines

Use the **set** form of this command to set the minimum interval between BGP routing advertisements to a BGP neighbor.

Use the **delete** form of this command to restore the advertisement interval to the default.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> allowas-in

Allows or disallows receiving BGP advertisements containing the AS path of the local router.

Syntax

set protocols bgp *asn* **neighbor** *id* **allowas-in** [**number** *num*]

delete protocols bgp *asn* **neighbor** *id* **allowas-in**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         allowas-in {
            number u32
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
number *num*	Optional. Specifies the number of times to accept an AS path containing the readvertisement of the local router’s ASN. The range is 1 to 10. The default is 3.

Default

Readvertisement is disabled.

Usage Guidelines

Use the **set** form of this command to allow the router to accept BGP AS paths advertising the router’s own AS number.

This situation could indicate a routing loop, and by default such an update is dropped. However, you can use this command to direct the router to accept updates readvertising its AS number for a specified number of times.

Use the **delete** form of this command to disallow readvertisement of the router’s AS path.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> attribute-unchanged

Allows the router to send updates to a neighbor with unchanged attributes.

Syntax

set protocols bgp *asn* **neighbor** *id* **attribute-unchanged** [**as-path** | **med** | **next-hop**]

delete protocols bgp *asn* **neighbor** *id* [**as-path** | **med** | **next-hop**]

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         attribute-unchanged {
            as-path
            med
            next-hop
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
as-path	Optional. Propagates the route update with unchanged AS_PATH attribute.
med	Optional. Propagates the route update with unchanged Multi Exit Discriminator.

next-hop	Optional. Propagates the route update with unchanged next hop.

Default

Disabled.

Usage Guidelines

Use the **set** form of this command to direct the router to propagate routing updates without modifying the BGP AS_PATH, Multi Exit Discriminator (MED), or next-hop attribute.

Use the **delete** form of this command to restore normal modification of BGP attributes.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> capability dynamic

Advertises support for dynamic update of BGP capabilities advertised and received from this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **capability dynamic**

delete protocols bgp *asn* **neighbor** *id* **capability dynamic**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         capability {
            dynamic
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.

Default

By default, the session is brought up with minimal capability on both sides.

Usage Guidelines

In general, BGP-4 requires that if a BGP speaker receives an OPEN message with an unrecognized optional parameters, the speaker must terminate BGP peering. This makes it difficult to introduce new capabilities into the protocol. The Capabilities parameter allows graceful negotiation of BGP capabilities without requiring a peer to terminate peering.

This command specifies use of the BGP Dynamic Capability feature, which allows dynamic update of capabilities over an established BGP session.

Use the **set** form of this command to specify dynamic update of BGP capabilities to be used in capability negotiation.

Use the **delete** form of this command to remove the dynamic update capability.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> capability orf

Advertises support for Outbound Route Filtering (ORF) for updating BGP capabilities advertised and received from this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **capability orf** [**prefix-list** [**receive** | **send**]]

delete protocols bgp *asn* **neighbor** *id* **capability orf**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    bgp asn {
        neighbor [ipv4|text] {
            capability {
                orf {
                    prefix-list {
                        receive
                        send
                    }
                }
            }
        }
    }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
prefix-list	Optional. Advertise prefix-list ORF capability to this neighbor.
receive	Optional. Advertise the ability to receive the ORF from this neighbor.

send	Optional. Advertise the capability to send the ORF to this neighbor.

Default

By default, the session is brought up with minimal capability on both sides.

Usage Guidelines

In general, BGP-4 requires that if a BGP speaker receives an OPEN message with an unrecognized optional parameters, the speaker must terminate BGP peering. This makes it difficult to introduce new capabilities into the protocol. The Capabilities parameter allows graceful negotiation of BGP capabilities without requiring a peer to terminate peering.

This command specifies use of BGP Outbound Route Filtering (ORF) to send and receive capabilities. Using ORF minimizes the number of BGP updates that are sent between peer routers.

Use the **set** form of this command to specify BGP ORF capabilities to be used in capability negotiation.

Use the **delete** form of this command to remove the ORF capability.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> default-originate

Enables or disables forwarding of the default route to a BGP neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **default-originate** [**route-map** *map-name*]

delete protocols bgp *asn* **neighbor** *id* **default-originate** [**route-map** *map-name*]

show protocols bgp *asn* **neighbor** *id* **default-originate**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         default-originate {
            route-map: text
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
route-map *map-name*	Optional. Specifies a configured route map to be used when advertising the default route.

Default

The default route is not distributed.

Usage Guidelines

Use the **set** form of this command to enable the router to advertise the default route 0.0.0.0 to this BGP neighbor. This route can then be used by the neighbor to reach the local router if no other routes are available.

The route 0.0.0.0 need not be explicitly configured on the local router.

If a route map is specified, the default route is advertised if two conditions are satisfied:

- The route map includes a **match ip** address clause.
- A route exists that exactly matches the IP access list.

Route maps are configured using the **protocols bgp <asn> neighbor <id> local-as** command (see page 215).

Use the **delete** form of this command to disable forwarding of the default route or to delete a route map.

Use the **show** form of this command to view BGP neighbor default route configuration settings.

protocols bgp <asn> neighbor <id> description <desc>

Provides a brief description for a BGP neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **description** *desc*

delete protocols bgp *asn* **neighbor** *id* **description**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    bgp asn {
        neighbor [ipv4|text] {
            description: text
        }
    }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
desc	Mandatory. A description of up to 80 characters for the neighbor. If spaces are included, the description must be enclosed in quotes.

Default

None.

Usage Guidelines

Use the **set** form of this command to enter a description for this BGP neighbor.

Use the **delete** form of this command to delete the neighbor’s description.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> disable-capability-negotiation

Disables BGP capability negotiation.

Syntax

set protocols bgp *asn* **neighbor** *id* **disable-capability-negotiation**

delete protocols bgp *asn* **neighbor** *id* **disable-capability-negotiation**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         disable-capability-negotiation
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.

Default

Capability negotiation is performed.

Usage Guidelines

Use the **set** form of this command to disable BGP capability negotiation.

Use the **delete** form of this command to delete this attribute and restore BGP capability negotiation.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> disable-connected-check

Disables direct connection verification for single-hop eBGP peers.

Syntax

set protocols bgp *asn* **neighbor** *id* **disable-connected-check**

delete protocols bgp *asn* **neighbor** *id* **disable-connected-check**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         disable-connected-check
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.

Default

Connection verification is performed.

Usage Guidelines

Use the **set** form of this command to disable connection verification.

Connection verification is used to establish an eBGP peering session with a single-hop peer. A single-hop peer is defined using the **neighbor ebgp-multihop** command and specifying a time-to-live (TTL) value of 1. Such a peer is reachable by a single hop, but is configured on a loopback interface or is otherwise configured with a non–directly connected IP address.

For these peers, the BGP process normally checks to determine whether the eBGP peer is directly connected to the same network as the local router. If not, the peering session is not established.

If connection verification is disabled, source updating must be enabled using the **protocols bgp <asn> neighbor <id> update-source <addr>** command (see page 253) so that the BGP process can used the loopback interface for the peering session.

Use the **delete** form of this command to restore connection verification for eBGP peering sessions.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> disable-send-community

Disables sending of community attributes to the specified neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **disable-send-community** [**extended** | **standard**]

delete protocols bgp *asn* **neighbor** *id* **disable-send-community**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         disable-send-community {
            extended
            standard
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
extended	Optional. Disable sending extended community attributes.
standard	Optional. Disable sending standard community attributes.

Default

None.

Usage Guidelines

Use this command to enable and disable sending community attributes between BGP neighbors.

Use the **set** form of this command to disable sending community attributes.

Use the **delete** form of this command to restore sending community attributes.

Use the show form of this command to view BGP neighbor configuration settings

protocols bgp <asn> neighbor <id> distribute-list export <acl-num>

Applies an access list to filter outbound routing updates to this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **distribute-list export** *acl-num*

delete protocols bgp *asn* **neighbor** *id* **distribute-list**

show protocols bgp *asn* **neighbor** *id* **distribute-list**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         distribute-list {
            export: 1-199
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
acl-num	Optional. The number of a standard or extended access list. The range for a standard access list is 1 to 99. The range for an extended access list is 100 to 199.

Default

None.

Usage Guidelines

Use the **set** form of this command to apply an access list to filter outbound routing updates to a BGP neighbor.

Use the **delete** form of this command to disable outbound distribute list filtering.

Use the **show** form of this command to view BGP neighbor distribute list configuration settings.

NOTE *A neighbor distribute list cannot be used together with a neighbor prefix list in the same direction. These two lists are mutually exclusive, and only one list may be applied to a given direction.*

protocols bgp <asn> neighbor <id> distribute-list import <acl-num>

Applies an access list to filter inbound routing updates from this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **distribute-list import** *acl-num*

delete protocols bgp *asn* **neighbor** *id* **distribute-list**

show protocols bgp *asn* **neighbor** *id* **distribute-list**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         distribute-list {
            import: 1-199
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
acl-num	Optional. The number of a standard or extended access list. The range for a standard access list is 1 to 99. The range for an extended access list is 100 to 199.

Default

None.

Usage Guidelines

Use the **set** form of this command to apply an access list to filter inbound routing updates from a BGP neighbor.

Use the **delete** form of this command to disable inbound distribute list filtering.

Use the **show** form of this command to view BGP neighbor distribute list configuration settings.

NOTE *A neighbor distribute list cannot be used together with a neighbor prefix list in the same direction. These two lists are mutually exclusive, and only one list may be applied to the specified direction.*

protocols bgp <asn> neighbor <id> ebgp-multihop <ttl>

Allows eBGP neighbors not on directly connected networks.

Syntax

set protocols bgp *asn* **neighbor** *id* **ebgp-multihop** *ttl*

delete protocols bgp *asn* **neighbor** *id* **ebgp-multihop**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         ebgp-multihop: 1-255
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
ttl	Mandatory. The time-to-live, or maximum number of hops allowed. The range is 1 to 255. The default is 10.

Default

Only directly connected neighbors are allowed.

Usage Guidelines

Use the **set** form of this command to allow connections to eBGP peers residing on networks that are not directly connected.

Use the **delete** form of this command to restrict connections to directly connected peers.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> filter-list export <acl-num>

Applies a filter list to routing updates to this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **filter-list export** *acl-num*

delete protocols bgp *asn* **neighbor** *id* **filter-list export** *acl-num*

show protocols bgp *asn* **neighbor** *id* **filter-list**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         filter-list {
            export: 1-65535
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
acl-num	Mandatory. Number of an autonomous system path access list. The range is 1 to 65535.

Default

None.

Usage Guidelines

Use the **set** form of this command to apply an access list for filtering outbound routing updates.

Use the **delete** form of this command to disable outbound routing update filtering.

Use the **show** form of this command to view BGP neighbor filter list configuration settings.

protocols bgp <asn> neighbor <id> filter-list import <acl-num>

Applies a filter list to routing updates from this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **filter-list import** *acl-num*

delete protocols bgp *asn* **neighbor** *id* **filter-list import** *acl-num*

show protocols bgp *asn* **neighbor** *id* **filter-list**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         filter-list {
            import: 1-65535
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
acl-num	Mandatory. Number of an autonomous system path access list. The range is 1 to 65535.

Default

None.

Usage Guidelines

Use the **set** form of this command to apply an access list for filtering inbound routing updates.

Use the **delete** form of this command to disable inbound routing update filtering.

Use the **show** form of this command to view BGP neighbor filter list configuration settings.

protocols bgp <asn> neighbor <id> local-as

Defines a local autonomous system number for eBGP peerings.

Syntax

set protocols bgp *asn* **neighbor** *id* **local-as** [**local-as** *asn* | **no-prepend**]

delete protocols bgp *asn* **neighbor** *id* **local-as** [**local-as** *asn* | **no-prepend**]

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         local-as {
            local-as: 1-65535
            no-prepend
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
asn	Optional. Valid autonomous system number. This may not be the number of the autonomous system to which the neighbor belongs. The range is 1 to 65535.
no-prepend	Optional. Directs the router to not prepend the local autonomous system number to routes received from an external peer.

Default

None.

Usage Guidelines

Use the **set** form of this command to set a local autonomous system number for eBGP peer groupings. This number is used by all peers in the group for peering. It cannot be applied to individual peers in the group.

A local autonomous system number can only be applied to a true eBGP peer; it cannot be applied to peers in different subautonomous systems within a confederation.

The **no-prepend** keyword can cause routing loops and should be used with care. It should be used only to change the autonomous system number in a BGP network. After the network transition has completed, this setting should be deleted.

Use the **delete** form of this command to remove a local autonomous system number, or to remove the **no-prepend** keyword.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> maximum-prefix <max-num>

Sets the maximum number of prefixes to accept from this neighbor before that neighbor is taken down.

Syntax

set protocols bgp *asn* **neighbor** *id* **maximum-prefix** *max-num*

delete protocols bgp *asn* **neighbor** *id* **maximum-prefix** *max-num*

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         maximum-prefix: u32
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
max-num	Mandatory. The maximum number of prefixes to accept from this neighbor before the neighbor is taken down.

Default

No maximum to the number of prefixes accepted.

Usage Guidelines

Use the **set** form of this command to set the maximum number of prefixes to accept from this neighbor before the neighbor is taken down. This helps alleviate situations where the router receives more routes than it can handle.

Use the **delete** form of this command to remove the maximum.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> nexthop-self

Sets the local router as the next hop for this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **nexthop-self**

delete protocols bgp *asn* **neighbor** *id* **nexthop-self**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         nexthop-self
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.

Default

Disabled.

Usage Guidelines

Use the **set** form of this command to set this router as the next hop for this neighbor. This disables the next-hop calculation for this neighbor.

Use the **delete** form of this command to restore next-hop calculation for the neighbor.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> override-capability

Overrides capability negotiation to allow a peering session to be established with a neighbor that does not support capability negotiation.

Syntax

set protocols bgp *asn* **neighbor** *id* **override-capability**

delete protocols bgp *asn* **neighbor** *id* **override-capability**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         override-capability
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.

Default

A session cannot be established if the neighbor does not support capability negotiation.

Usage Guidelines

Use the **set** form of this command to enable a peering session to be established with a BGP neighbor that does not support capability negotiation.

Normally, if a BGP peer does not support capability negotiation, a peering session cannot be established and the connection is terminated. Setting this value overrides this process and allows the session to be established.

Use the **delete** form of this command to restore the default behavior.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> passive

Directs the router not to initiate connections with this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **passive**

delete protocols bgp *asn* **neighbor** *id* **passive**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         passive
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.

Default

The router both accepts inbound connections and initiates outbound connections.

Usage Guidelines

Use the **set** form of this command to configure the local router such that accepts inbound connections from the neighbor, but does not initiate outbound connections by sending a BGP OPEN message.

Use the **delete** form of this command to restore the default behavior.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> password <pwd>

Specifies the BGP MD5 password.

Syntax

set protocols bgp *asn* **neighbor** *id* **password** *pwd*

delete protocols bgp *asn* **neighbor** *id* **password** *pwd*

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         password text
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
pwd	Mandatory. The password used to generate the MD5 digest used for verification by the BGP neighbor. If the neighbor's generated MD5 digest does not match the originated MD5 digest, any communication is silently discarded.

Default

None.

Usage Guidelines

Use the **set** form of this command to specify the BGP MD5 password.

Use the **delete** form of this command to remove the BGP MD5 password.

Use the **show** form of this command to view the BGP MD5 password.

protocols bgp <asn> neighbor <id> peer-group

Creates a peer group, or assigns a neighbor as a member of the specified peer group.

Syntax

To create a BGP peer group, the syntax is as follows:

set protocols bgp *asn* **neighbor** *group-name* **peer-group**

delete protocols bgp *asn* **neighbor** *group-name* **peer-group**

show protocols bgp *asn* **neighbor** *group-name*

To add a BGP neighbor to a peer group, the syntax is as follows:

set protocols bgp *asn* **neighbor** *ipv4* **peer-group** *group-name*

delete protocols bgp *asn* **neighbor** *ipv4* **peer-group** *group-name*

show protocols bgp *asn* **neighbor** *ipv4*

Command Mode

Configuration mode.

Configuration Statement

The configuration statement for a BGP peer group is as follows:

```
protocols {
   bgp asn {
      neighbor text{
         peer-group
      }
   }
}
```

The configuration statement for a BGP neighbor that is a member of a BGP peer group is as follows:

```
protocols {
   bgp asn {
      neighbor ipv4{
         peer-group: text
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
ipv4	Mandatory when assigning a neighbor to a peer group. The IPv4 address of a defined BGP neighbor.
group-name	Mandatory. A string identifying a BGP peer group.

Default

By default, peer group members inherit the following settings, if configured:

- Remote AS
- Update source
- Outbound route maps
- Outbound filter lists
- Outbound distribute lists
- Nexthop-self settings
- BGP Peer Groups

Usage Guidelines

Use the **set** form of this command to define a peer group or to assign this BGP neighbor to a peer-group.

Configuring a peer group simplifies configuration for neighbors with similar update policies. Once a peer group is created, it can be configured using the same commands for configuring individual BGP neighbors. Each member of the peer group inherits the peer group’s update policies unless overridden by explicit configuration of the individual peer. This also makes update calculation more efficient.

All members of a peer group must share identical outbound routing policies—that is, they must have identified distribute lists, filter lists, prefix lists, and route maps applied. They need not have identical settings for default-originate, as this is always processed on a per-peer basis. Inbound update routing policies may vary per peer group member.

Use the **delete** form of this command to remove a peer group or to remove a neighbor from a peer group.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> port <port-num>

Specifies the port on which the neighbor is listening for BGP signals.

Syntax

set protocols bgp *asn* **neighbor** *id* **port** *port-num*

delete protocols bgp *asn* **neighbor** *id* **port**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         port: 1-65535
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
port-num	Mandatory. The port on which the BGP neighbor will be listening for BGP messages. The range is 1 to 65535. The default is 179.

Default

By default, the router uses the well-known port for BGP, which is 179.

Usage Guidelines

Use the **set** form of this command to specify the port number to which BGP signals will be sent.

Use the **delete** form of this command to restore the port to the default.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> prefix-list export <list-name>

Applies an prefix list to filter updates to this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **prefix-list export** *list-name*

delete protocols bgp *asn* **neighbor** *id* **prefix-list export** *list-name*

show protocols bgp *asn* **neighbor** *id* **prefix-list**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         prefix-list {
            export: text
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
list-name	Mandatory. Name of a configured prefix list.

Default

None.

Usage Guidelines

Use the **set** form of this command to restrict distribution of outbound BGP neighbor information by filtering with a prefix list.

Use the **delete** form of this command to remove an outbound prefix list filter.

Use the **show** form of this command to view BGP neighbor prefix list configuration settings.

protocols bgp <asn> neighbor <id> prefix-list import <list-name>

Applies an prefix list to filter updates from this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **prefix-list import** *list-name*

delete protocols bgp *asn* **neighbor** *id* **prefix-list import** *list-name*

show protocols bgp *asn* **neighbor** *id* **prefix-list**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         prefix-list {
            import: text
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
list-name	Mandatory. Name of a configured prefix list.

Default

None.

Usage Guidelines

Use the **set** form of this command to restrict distribution of inbound BGP neighbor information by filtering with a prefix list.

Use the **delete** form of this command to remove an inbound prefix list filter.

Use the **show** form of this command to view BGP neighbor prefix list configuration settings.

protocols bgp <asn> neighbor <id> remote-as <asn>

Specifies the autonomous system number of the neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **remote-as** *asn*

delete protocols bgp *asn* **neighbor** *id* **remote-as**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         remote-as: u32
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.

Default

None.

Usage Guidelines

Use the **set** form of this command to specify the autonomous system (AS) number of a BGP neighbor.

If the AS number of the neighbor is the same as that of the local router, the neighbor is an internal BGP (iBGP) peer. If it is different the neighbor is an external BGP (eBGP) peer.

Use the **delete** form of this command to remove AS number settings for the neighbor.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> remove-private-as

Directs the router to remove private AS numbers from updates sent to this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **remove-private-as**

delete protocols bgp *asn* **neighbor** *id* **remove-private-as**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         remove-private-as
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.

Default

Private AS numbers are included in outgoing updates.

Usage Guidelines

Use the **set** form of this command to direct the router to exclude private autonomous system (AS) numbers from updates to eBGP peers. When this feature is enabled, the router omits private AS numbers from the AS_PATH attribute. The range of private AS numbers is 64512 to 65535.

Note that it is a configuration error to include both private and public AS numbers in an AS path. If the router detects this error, it does not remove private AS numbers.

This command may be used in confederations provided that the private AS numbers are appended after the confederation portion of the AS path.

This command applies only to eBGP peers; it cannot be used with iBGP peers.

Use the **delete** form of this command to restore the default behavior.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> route-map export <map-name>

Applies a route map to filter updates to this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **route-map export** *map-name*

delete protocols bgp *asn* **neighbor** *id* **route-map export** *map-name*

show protocols bgp *asn* **neighbor** *id* **route-map export** *map-name*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         route-map {
            export: text
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
map-name	Mandatory. Name of a configured route map.

Default

None.

Usage Guidelines

Use the **set** form of this command to restrict distribution of outbound BGP neighbor information by filtering with a route map.

Use the **delete** form of this command to remove an outbound route map filter.

Use the **show** form of this command to view BGP neighbor route map configuration settings.

protocols bgp <asn> neighbor <id> route-map import <map-name>

Applies a route map to filter updates to or from this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **route-map import** *map-name*

delete protocols bgp *asn* **neighbor** *id* **route-map import** *map-name*

show protocols bgp *asn* **neighbor** *id* **route-map import** *map-name*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         route-map {
            import: text
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
map-name	Mandatory. Name of a configured route map.

Default

None.

Usage Guidelines

Use the **set** form of this command to restrict distribution of BGP neighbor information by filtering with a route map.

Use the **delete** form of this command to remove a route map filter.

Use the **show** form of this command to view BGP neighbor route map configuration settings.

protocols bgp <asn> neighbor <id> shutdown

Administratively shuts down a BGP neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **shutdown**

delete protocols bgp *asn* **neighbor** *id* **shutdown**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         shutdown
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.

Default

Disabled.

Usage Guidelines

Use the **set** form of this command to administratively shut down a BGP neighbor.

Shutting down a BGP speaker administratively terminates any active sessions for the neighbor and removes any associated routing information. If the neighbor is a peer group, this could terminate a large number of sessions.

To determine whether a BGP neighbor has been administratively shut down, you can view BGP summary information. A neighbor with a status of Idle with a comment of Admin have been administratively shut down.

Use the **delete** form of this command to administratively reenable a BGP neighbor.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> soft-reconfiguration

Directs the router to store received routing updates.

Syntax

set protocols bgp *asn* **neighbor** *id* **soft-reconfiguration** [**inbound**]

delete protocols bgp *asn* **neighbor** *id* **soft-reconfiguration**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         soft-reconfiguration {
            inbound
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
inbound	Optional. Indicates that the update to be stored is an incoming update.

Default

Disabled.

Usage Guidelines

Use the **set** form of this command to enable soft reconfiguration.

When this command is issued, the router begins storing routing updates, which can be used subsequently for inbound soft reconfiguration. Outbound BGP soft reconfiguration can be performed without enabling inbound soft reconfiguration.

Soft reconfiguration is memory-intensive and is not recommended for long periods of time.

Use the **delete** form of this command to disable soft reconfiguration.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> strict-capability-match

Directs the router to strictly match the capabilities of the neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **strict-capability-match**

delete protocols bgp *asn* **neighbor** *id* **strict-capability-match**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         strict-capability-match
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.

Default

Disabled.

Usage Guidelines

Use the **set** form of this command to direct the router strictly compare the capabilities of the local router and the remote peer. If the capabilities are different, the session is terminated.

Use the **delete** form of this command to disable strict capability matching.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> timers

Sets BGP timers for this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **timers** [**connect** *seconds* | **keepalive** *seconds* | **holdtime** *seconds*]

delete protocols bgp *asn* **neighbor** *id* **timers** [**connect** | **keepalive** | **holdtime**]

show protocols bgp *asn* **neighbor** *id* **timers**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         timers {
            connect: 0-65535
            keepalive: 1-65535
            holdtime: 0, 4-65535
         }
      }
   }
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
connect *seconds*	Optional. The amount of time, in seconds, that the system waits for the transport protocol connection to complete. If this timer expires, the state remains in Connect state, the timer is reset, and the system tries to initiate a new transport connection. The range is 0 to 65535.The default is 120. During the connect period, the remote BGP peer can establish a connection to the local system.

keepalive *seconds*	Optional. The frequency, in seconds, with which the local router sends keep-alive messages to this neighbor. The range is 1 to 65535, where 0 disables the keep-alive timer. The default is 60.
holdtime *seconds*	Optional. The maximum interval, in seconds, after which if the local router has not received a keep-alive message from this neighbor, the neighbor is declared dead. The range is 0 and 4 to 65535, where 0 disables the holdtime timer. The default is 180.

Default

The default for the connect timer is 120. The default for the keep-alive timer is 60 seconds. The default for the holdtime timer is 180 seconds.

Usage Guidelines

Use the **set** form of this command to set timers for monitoring the health of the remote peer.

- If this command is issued and values are specified, the values configured here override global timers set for the local router.
- If this command is issued but values are not specified, the defaults for this command apply.
- If this command is not issued, or if the configuration statement is deleted, timers set globally for the router using the **protocols bgp <asn> timers** command (see page 121) apply to this neighbor.

Use the **delete** form of this command to remove explicitly configured timers for a neighbor. In this case, timers set globally for the router apply to this neighbor.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> unsuppress-map <map-name>

Directs the router to selectively advertise routes suppressed by aggregating addresses, based on a route map.

Syntax

set protocols bgp *asn* **neighbor** *id* **unsuppress-map** *map-name*

delete protocols bgp *asn* **neighbor** *id* **unsuppress-map**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         unsuppress-map
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
map-name	Mandatory. The name of a configured route map.

Default

Routes suppressed by address aggregation are not advertised.

Usage Guidelines

Use the **set** form of this command to selectively advertise routes suppressed by aggregating addresses.

Use the **delete** form of this command to restore the default behavior.

Use the **show** form of this command to view BGP neighbor configuration settings.

protocols bgp <asn> neighbor <id> update-source <addr>

Specifies the source IP address for routing updates.

Syntax

set protocols bgp *asn* **neighbor** *id* **update-source** *addr*

delete protocols bgp *asn* **neighbor** *id* **update-source**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         update-source ipv4
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
addr	Mandatory. The IPv4 address of the router to receive routing updates from.

Default

None.

Usage Guidelines

Use the **set** form of this command to direct the system to use a specific router for routing updates.

Use the **delete** form of this command to remove the routing update source.

Use the **show** form of this command to view the configuration settings.

protocols bgp <asn> neighbor <id> weight <weight>

Defines a default weight for routes from this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *id* **weight** *weight*

delete protocols bgp *asn* **neighbor** *id* **weight**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         weight: 0-65535
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory. The IP address of a BGP neighbor or the name of a BGP peer group.
weight	Mandatory. The weight to be assigned to routes from this neighbor. The range is 0 to 65535.

Default

Routes learned from a BGP neighbor have a weight of 0. Routes sourced by the local router have a weight of 32768.

Usage Guidelines

Use the **set** form of this command to set the default weights for routes learned from a BGP neighbor.

Use the **delete** form of this command to restore route weighting to the default. Existing route weights are not changed.

Use the **show** form of this command to view BGP neighbor configuration settings.

show ip bgp ipv4 unicast neighbors

Displays information for IPv4 unicast BGP neighbors.

Syntax

show ip bgp ipv4 unicast neighbors [*ipv4*]

Command Mode

Operational mode.

Parameters

ipv4	Displays detailed information for the specified IPv4 unicast BGP neighbor.

Default

Information is shown for all IPv4 unicast BGP neighbors.

Usage Guidelines

Use this command to display BGP IPv4 unicast neighbor information.

show ip bgp ipv4 unicast neighbors <ipv4> advertised-routes

Displays advertised BGP IPv4 unicast routes for a BGP neighbor.

Syntax

show ip bgp ipv4 unicast neighbors *ipv4* **advertised-routes**

Command Mode

Operational mode.

Parameters

ipv4	Mandatory. The IP address of an IPv4 unicast BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display advertised BGP IPv4 unicast routes for a BGP neighbor.

show ip bgp ipv4 unicast neighbors <ipv4> prefix-counts

Displays IPv4 unicast prefix-counts for a BGP neighbor.

Syntax

show ip bgp ipv4 unicast neighbors *ipv4* **prefix-counts**

Command Mode

Operational mode.

Parameters

ipv4	Mandatory. The IP address of an IPv4 unicast BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display IPv4 unicast prefix counts for a BGP neighbor.

show ip bgp ipv4 unicast neighbors <ipv4> received prefix-filter

Displays the IPv4 unicast prefix-lists received from a BGP neighbor.

Syntax

show ip bgp ipv4 unicast neighbors *ipv4* **received prefix-filter**

Command Mode

Operational mode.

Parameters

ipv4	Mandatory. The IP address of an IPv4 unicast BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display BGP IPv4 unicast prefix-lists received from a BGP neighbor.

show ip bgp ipv4 unicast neighbors <ipv4> received-routes

Displays the IPv4 unicast routes received from a BGP neighbor.

Syntax

show ip bgp ipv4 unicast neighbors *ipv4* **received-routes**

Command Mode

Operational mode.

Parameters

ipv4	Mandatory. The IP address of an IPv4 unicast BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display IPv4 unicast routes (both accepted and rejected) received from a BGP neighbor.

show ip bgp ipv4 unicast neighbors <ipv4> routes

Displays IPv4 unicast received and accepted routes from a BGP neighbor.

Syntax

show ip bgp ipv4 unicast neighbors *ipv4* **routes**

Command Mode

Operational mode.

Parameters

ipv4	Mandatory. The IP address of an IPv4 unicast BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display IPv4 unicast received and accepted routes from a BGP neighbor.

show ip bgp neighbors

Displays BGP neighbor information.

Syntax

show ip bgp neighbors [*ipv4*]

Command Mode

Operational mode.

Parameters

ipv4	Optional. The IP address of a BGP neighbor.

Default

Information is shown for all BGP neighbors.

Usage Guidelines

Use this command to display BGP neighbor information.

show ip bgp neighbors <ipv4> advertised-routes

Displays advertised routes for a BGP neighbor.

Syntax

show ip bgp neighbors *ipv4* **advertised-routes**

Command Mode

Operational mode.

Parameters

ipv4	Mandatory. The IP address of a BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display advertised routes for a BGP neighbor.

show ip bgp neighbors <ipv4> dampened-routes

Displays dampened routes to a BGP neighbor.

Syntax

show ip bgp neighbors *ipv4* **dampened-routes**

Command Mode

Operational mode.

Parameters

ipv4	Mandatory. The IP address of a BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display routes that have been dampened (suppressed) to a BGP neighbor due to route flapping.

show ip bgp neighbors <ipv4> flap-statistics

Displays route flap statistics for routes learned from a BGP neighbor.

Syntax

show ip bgp neighbors *ipv4* **flap-statistics**

Command Mode

Operational mode.

Parameters

ipv4	Mandatory. The IP address of a BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display route flap statistics for routes learned from a BGP neighbor.

show ip bgp neighbors <ipv4> prefix-counts

Displays prefix counts for a BGP neighbor.

Syntax

show ip bgp neighbors *ipv4* **prefix-counts**

Command Mode

Operational mode.

Parameters

ipv4	Mandatory. The IP address of a BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display prefix counts for a BGP neighbor.

show ip bgp neighbors <ipv4> received prefix-filter

Displays prefix lists received from a BGP neighbor.

Syntax

show ip bgp neighbors *ipv4* **received prefix-filter**

Command Mode

Operational mode.

Parameters

ipv4	Mandatory. The IP address of a BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display prefix lists received from a BGP neighbor.

show ip bgp neighbors <ipv4> received-routes

Displays routes received from a BGP neighbor.

Syntax

show ip bgp neighbors *ipv4* **received-routes**

Command Mode

Operational mode.

Parameters

ipv4	Mandatory. The IP address of a BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display routes (both accepted and rejected) received from a BGP neighbor.

show ip bgp neighbors <ipv4> routes

Displays all received and accepted routes from a BGP neighbor.

Syntax

show ip bgp neighbors *ipv4* **routes**

Command Mode

Operational mode.

Parameters

ipv4	Mandatory. The IP address of a BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display received and accepted routes from a BGP neighbor.

Chapter 6: Route Redistribution

This chapter describes commands for BGP route redistribution.

This chapter presents the following topics:

- Route Redistribution Commands

Route Redistribution Commands

This chapter contains the following commands.

Configuration Commands		
protocols bgp <asn> redistribute connected	Configuration	Redistributes directly connected routes into BGP.
protocols bgp <asn> redistribute kernel	Configuration	Redistributes kernel routes into BGP.
protocols bgp <asn> redistribute ospf	Configuration	Redistributes routes learned from OSPF into BGP.
protocols bgp <asn> redistribute rip	Configuration	Redistributes routes learned from RIP into BGP.
protocols bgp <asn> redistribute static	Configuration	Redistributes static routes into BGP.
Operational Commands		
None.		

protocols bgp <asn> redistribute connected

Redistributes directly connected routes into BGP.

Syntax

set protocols bgp asn **redistribute connected** [**metric** *metric* | **route-map** *map-name*]

delete protocols bgp *asn* **redistribute connected** [**metric** | **route-map**]

show protocols bgp *asn* **redistribute**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      redistribute {
         connected {
            metric: u32
            route-map: text
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
metric	Optional. The metric to be applied to redistributed connected routes.
map-name	Optional. The name of a configured route map to be used for redistributing connected routes.

Default

When this command has not been set, directly connected routes are not distributed into BGP.

Usage Guidelines

Use this command to redistribute directly connected routes into BGP.

Use the **set** form of this command to direct the router to redistribute directly connected routes into BGP.

Use the **delete** form of this command to prevent redistribution of directly connected routes into BGP.

Use the **show** form of this command to view route redistribution configuration settings.

protocols bgp <asn> redistribute kernel

Redistributes kernel routes into BGP.

Syntax

set protocols bgp asn **redistribute kernel** [**metric** *metric* | **route-map** *map-name*]

delete protocols bgp *asn* **redistribute kernel** [**metric** | **route-map**]

show protocols bgp *asn* **redistribute**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      redistribute {
         kernel {
            metric: u32
            route-map: text
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
metric	Optional. The metric to be applied to redistributed kernel routes.
map-name	Optional. The name of a configured route map to be used for redistributing kernel routes.

Default

When this command has not been set, kernel routes are not distributed into BGP.

Usage Guidelines

Use this command to redistribute kernel routes into BGP.

Use the **set** form of this command to direct the router to redistribute kernel routes into BGP.

Use the **delete** form of this command to prevent redistribution of kernel routes into BGP.

Use the **show** form of this command to view route redistribution configuration settings.

protocols bgp <asn> redistribute ospf

Redistributes routes learned from OSPF into BGP.

Syntax

set protocols bgp asn **redistribute ospf** [**metric** *metric* | **route-map** *map-name*]

delete protocols bgp *asn* **redistribute ospf** [**metric** | **route-map**]

show protocols bgp *asn* **redistribute**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      redistribute {
         ospf {
            metric: u32
            route-map: text
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
metric	Optional. The metric to be applied to redistributed OSPF routes.
map-name	Optional. The name of a configured route map to be used for redistributing OSPF routes.

Default

When this command has not been set, routes learned from OSPF are not distributed into BGP.

Usage Guidelines

Use this command to redistribute Open Shortest Path First (OSPF) routes into BGP.

Use the **set** form of this command to direct the router to redistribute routes learned from OSPF into BGP.

Use the **delete** form of this command to prevent redistribution of routes learned from OSPF into BGP.

Use the **show** form of this command to view route redistribution configuration settings.

protocols bgp <asn> redistribute rip

Redistributes routes learned from RIP into BGP.

Syntax

set protocols bgp asn **redistribute rip** [**metric** *metric* | **route-map** *map-name*]

delete protocols bgp *asn* **redistribute rip** [**metric** | **route-map**]

show protocols bgp *asn* **redistribute**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      redistribute {
         rip {
            metric: u32
            route-map: text
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
metric	Optional. The metric to be applied to redistributed RIP routes.
map-name	Optional. The name of a configured route map to be used for redistributing RIP routes.

Default

When this command has not been set, routes learned from RIP are not distributed into BGP.

Usage Guidelines

Use this command to redistribute Routing Information Protocol (RIP) routes into BGP.

Use the **set** form of this command to direct the router to redistribute routes learned from RIP into BGP.

Use the **delete** form of this command to prevent redistribution of routes learned from RIP into BGP.

Use the **show** form of this command to view route redistribution configuration settings.

protocols bgp <asn> redistribute static

Redistributes static routes into BGP.

Syntax

set protocols bgp asn **redistribute static** [**metric** *metric* | **route-map** *map-name*]

delete protocols bgp *asn* **redistribute static** [**metric** | **route-map**]

show protocols bgp *asn* **redistribute**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      redistribute {
         static {
            metric: u32
            route-map: text
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
metric	Optional. The metric to be applied to redistributed static routes.
map-name	Optional. The name of a configured route map to be used for redistributing static routes.

Default

When this command has not been set static routes are not distributed into BGP.

Usage Guidelines

Use this command to redistribute static routes into BGP.

Use the **set** form of this command to direct the router to redistribute static routes into BGP.

Use the **delete** form of this command to prevent redistribution of static routes into BGP.

Use the **show** form of this command to view route redistribution configuration settings.

Chapter 7: Route Server

This chapter describes commands for BGP route server.

This chapter presents the following topics:

- Route Server Commands

Route Server Commands

This chapter contains the following commands.

Configuration Commands		
protocols bgp <asn> neighbor <id> route-server-client	Configuration	Defines the local router as a BGP route server, with the neighbor as a route server client.
Operational Commands		
clear ip bgp view <view-name> <address>	Operational	Resets BGP connections for a view in a BGP route server.
clear ip bgp view <view-name> <address> ipv4 unicast	Operational	Resets IPv4 unicast BGP connections for a view in a BGP route server.
show ip bgp ipv4 unicast rsclient summary	Operational	Displays IPv4 unicast BGP route server client summary information.
show ip bgp rsclient <address>	Operational	Displays BGP route server client information.
show ip bgp view <view-name>	Operational	Displays BGP information for a view in a BGP route server.
show ip bgp view <view-name> ipv4 unicast rsclient summary	Operational	Displays IPv4 unicast route server client summary information for a view in a BGP route server.
show ip bgp view <view-name> ipv4 unicast summary	Operational	Displays IPv4 unicast summary information for a view in a BGP route server.
show ip bgp view <view-name> neighbors	Operational	Displays BGP neighbor information for a view in a BGP route server.
show ip bgp view <view-name> rsclient	Operational	Displays BGP route server client information for a view in a BGP route server.

clear ip bgp view <view-name> <address>

Resets BGP connections for a view in a BGP route server.

Syntax

clear ip bgp view *view-name* {*ipv4* | *ipv6*} [**in** [**prefix-filter**] | **soft** [**in** | **out**] | **rsclient**]

Command Mode

Operational mode.

Parameters

view-name	Mandatory. The name of a view (routing table) in a BGP route server.
ipv4	Resets the connection for the IPv4 BGP peer at the specified address.
ipv6	Resets the connection for the IPv6 BGP peer at the specified address.
in	Optional. Resets inbound sessions only.
prefix-filter	Optional. Clears the BGP outbound route filter (ORF). This keyword is ignored unless ORF capabilities have been enabled on the local system or received from the sending BGP peer. In this case, a normal inbound soft reset is performed.
soft	Optional. Uses soft reconfiguration for the reset.
in	Optional. Resets with soft reconfiguration only inbound sessions.
out	Optional. Resets with soft reconfiguration only outbound sessions.
rsclient	Optional. Resets only connections in the route server client Routing Information Base (RIB).

Default

When used without the **soft** option, reset connections are dropped, both inbound and outbound.

Usage Guidelines

Use this command to clear BGP connection statistics for a view. This forces BGP updates to be generated and new BGP policies to be applied.

Using a BGP route server is a way of solving the scalability problem that results from the requirement that iBGP peers be fully meshed. When a route server is employed, BGP routers peer only to the route server, and the route server servers BGP information to other BGP routers. This greatly reduces the number of BGP connections required.

Unlike a normal BGP router, a BGP router server must have several routing tables for managing the various routing policies of each BGP speakers: each of these routing tables is called a “view.”

Unless the **soft** option is used, all connections are dropped (a “hard reset”): TCP connections are terminated and all routes received from the neighbor are removed from the BGP routing table. Then the connection with the neighbor is reestablished.

If the **soft** option is used, routes from the neighbor are marked as stale but are not immediately removed from the BGP table. Stale routes that are not received from the neighbor when the connection is reestablished are removed from the BGP table at that point.

clear ip bgp view <view-name> <address> ipv4 unicast

Resets IPv4 unicast BGP connections for a view in a BGP route server.

Syntax

clear ip bgp view *view-name* {*ipv4* | *ipv6*} **ipv4 unicast** [**in** [**prefix-filter**] | **soft** [**in** | **out**]]

Command Mode

Operational mode.

Parameters

view-name	Mandatory. The name of a view (routing table) in a BGP route server for which connection statistics are to be cleared.
ipv4	Resets the connection for the IPv4 BGP peer at the specified address.
ipv6	Resets the connection for the IPv6 BGP peer at the specified address.
in	Optional. Resets inbound sessions only.
prefix-filter	Optional. Clears the BGP outbound route filter (ORF). This keyword is ignored unless ORF capabilities have been enabled on the local system or received from the sending BGP peer. In this case, a normal inbound soft reset is performed.
soft	Optional. Uses soft reconfiguration for the reset.
in	Optional. Resets with soft reconfiguration only inbound sessions.
out	Optional. Resets with soft reconfiguration only outbound sessions.
rsclient	Optional. Resets only connections in the route server client Routing Information Base (RIB).

Default

When used without the **soft** option, reset connections are dropped, both inbound and outbound.

Usage Guidelines

Use this command on a router running BGP to reset BGP sessions for a view (routing table) in a BGP route server. This forces BGP updates to be generated and new BGP policies to be applied.

Using a BGP route server is a way of solving the scalability problem that results from the requirement that iBGP peers be fully meshed. When a route server is employed, BGP routers peer only to the route server, and the route server servers BGP information to other BGP routers. This greatly reduces the number of BGP connections required.

Unlike a normal BGP router, a BGP router server must have several routing tables for managing the various routing policies of each BGP speakers: each of these routing tables is called a “view.”

Unless the **soft** option is used, all connections are dropped (a “hard reset”): TCP connections are terminated and all routes received from the neighbor are removed from the BGP routing table. Then the connection with the neighbor is reestablished.

If the **soft** option is used, routes from the neighbor are marked as stale but are not immediately removed from the BGP table. Stale routes that are not received from the neighbor when the connection is reestablished are removed from the BGP table at that point.

protocols bgp <asn> neighbor <id> route-server-client

Defines the local router as a BGP route server, with the neighbor as a route server client.

Syntax

set protocols bgp *asn* **neighbor** *id* **route-server-client**

delete protocols bgp *asn* **neighbor** *id* **route-server-client**

show protocols bgp *asn* **neighbor** *id*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor [ipv4|text] {
         route-server-client
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
id	Mandatory.The IP address of a BGP neighbor or the name of a BGP peer group.

Default

There is no route server in the autonomous system.

Usage Guidelines

Use the **set** form of this command to make the local router a BGP route server, and to designate the specified neighbor as a route server client.

Use the **delete** form of this command to remove the neighbor as a route server client.

Use the **show** form of this command to view BGP neighbor configuration settings.

show ip bgp ipv4 unicast rsclient summary

Displays IPv4 unicast BGP route server client summary information.

Syntax

show ip bgp ipv4 unicast rsclient summary

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display IPv4 unicast BGP route server client summary information.

show ip bgp rsclient <address>

Displays BGP route server client information.

Syntax

show ip bgp rsclient *address* [*ipv4* | *ipv4net* | **summary**]

Command Mode

Operational mode.

Parameters

ipv4	Optional. Displays routes for the route server client at the specified IPv4 address.
ipv4net	Optional. Displays routes for route server clients on the specified IPv4 network.
summary	Optional. Displays summary information for all route server clients.

Default

Detailed information is displayed for all route server clients.

Usage Guidelines

Use this command to display BGP route server client information.

show ip bgp view <view-name>

Displays BGP information for a view in a BGP route server.

Syntax

show ip bgp view *view-name* [*ipv4* | *ipv4net* | **summary**]

Command Mode

Operational mode.

Parameters

view-name	Mandatory. The name of a view (routing table) in a BGP route server.
ipv4	Optional. Show BGP information for the specified address.
ipv4net	Optional. Show BGP information for the specified network.
summary	Optional. Shows summary information for the specified view.

Default

None.

Usage Guidelines

Use this command to display BGP information for a view in a BGP route server.

Using a BGP route server is a way of solving the scalability problem that results from the requirement that iBGP peers be fully meshed. When a route server is employed, BGP routers peer only to the route server, and the route server servers BGP information to other BGP routers. This greatly reduces the number of BGP connections required.

Unlike a normal BGP router, a BGP router server must have several routing tables for managing the various routing policies of each BGP speakers: each of these routing tables is called a “view.”

show ip bgp view <view-name> ipv4 unicast rsclient summary

Displays IPv4 unicast route server client summary information for a view in a BGP route server.

Syntax

show ip bgp view *view-name* **ipv4 unicast rsclient summary**

Command Mode

Operational mode.

Parameters

view-name	Mandatory. The name of a view (routing table) in a BGP route server.

Default

None.

Usage Guidelines

Use this command to display BGP IPv4 unicast route server client summary information for a view in a BGP route server.

Using a BGP route server is a way of solving the scalability problem that results from the requirement that iBGP peers be fully meshed. When a route server is employed, BGP routers peer only to the route server, and the route server servers BGP information to other BGP routers. This greatly reduces the number of BGP connections required.

Unlike a normal BGP router, a BGP router server must have several routing tables for managing the various routing policies of each BGP speakers: each of these routing tables is called a “view.”

show ip bgp view <view-name> ipv4 unicast summary

Displays IPv4 unicast summary information for a view in a BGP route server.

Syntax

show ip bgp view *view-name* **ipv4 unicast summary**

Command Mode

Operational mode.

Parameters

view-name	Mandatory. The name of a view (routing table) in a BGP route server.

Default

None.

Usage Guidelines

Use this command to display BGP IPv4 unicast summary information for a view in a BGP route server.

Using a BGP route server is a way of solving the scalability problem that results from the requirement that iBGP peers be fully meshed. When a route server is employed, BGP routers peer only to the route server, and the route server servers BGP information to other BGP routers. This greatly reduces the number of BGP connections required.

Unlike a normal BGP router, a BGP router server must have several routing tables for managing the various routing policies of each BGP speakers: each of these routing tables is called a “view.”

show ip bgp view <view-name> neighbors

Displays BGP neighbor information for a view in a BGP route server.

Syntax

show ip bgp view *view-name* **neighbors** [*ipv4*]

Command Mode

Operational mode.

Parameters

view-name	Mandatory. The name of a view (routing table) in a BGP route server.
ipv4	Optional. Displays information about the neighbor at the specified IPv4 address.
ipv6	Optional. Displays information about the neighbor at the specified IPv6 address.

Default

Information is displayed for all BGP neighbors in the view.

Usage Guidelines

Use this command to display BGP neighbor information for a view in a BGP route server.

Using a BGP route server is a way of solving the scalability problem that results from the requirement that iBGP peers be fully meshed. When a route server is employed, BGP routers peer only to the route server, and the route server servers BGP information to other BGP routers. This greatly reduces the number of BGP connections required.

Unlike a normal BGP router, a BGP router server must have several routing tables for managing the various routing policies of each BGP speakers: each of these routing tables is called a "view."

show ip bgp view <view-name> rsclient

Displays BGP route server client information for a view in a BGP route server.

Syntax

show ip bgp view *view-name* **rsclient** {*ipv4* | *ipv6* | *ipv4net*| | **summary**}

Command Mode

Operational mode.

Parameters

view-name	Mandatory. The name of a view (routing table) in a BGP route server.
ipv4	Displays route server client information for the specified IPv4 neighbor.
ipv6	Displays route server client information for the specified IPv6 neighbor.
ipv4net	Displays information for all route server clients on the specified network.
summary	Shows summary route server client information.

Default

Detailed information is displayed.

Usage Guidelines

Use this command to display BGP route server client information for a view in a BGP route server.

Using a BGP route server is a way of solving the scalability problem that results from the requirement that iBGP peers be fully meshed. When a route server is employed, BGP routers peer only to the route server, and the route server servers BGP information to other BGP routers. This greatly reduces the number of BGP connections required.

Unlike a normal BGP router, a BGP router server must have several routing tables for managing the various routing policies of each BGP speakers: each of these routing tables is called a “view.”

Glossary of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AS	autonomous system
ARP	Address Resolution Protocol
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DLCI	data-link connection identifier
DMI	desktop management interface
DMZ	demilitarized zone
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EGP	Exterior Gateway Protocol

ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Ouput
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP security
IPv4	IP Version 4
IPv6	IP Version 6
ISP	Internet Service Provider
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
MAC	medium access control
MIB	Management Information Base
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit

NAT	Network Address Translation
ND	Neighbor Discovery
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PCI	peripheral component interconnect
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
Rx	receive
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical Network
SSH	Secure Shell
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus

TCP	Transmission Control Protocol
ToS	Type of Service
Tx	transmit
UDP	User Datagram Protocol
vif	virtual interface
VLAN	virtual LAN
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network