ADTRAN

6AOSCG0024-29D
December 2014

Configuration Guide

Configuring Border Gateway Protocol in AOS for Releases 18.03.00/R10.1.0 or Later

> NOTE
>
> *This guide only addresses BGP in AOS* ***data*** *products using AOS firmware* ***18.03.00*** *or later and AOS* ***voice*** *products using AOS firmware* ***R10.1.0*** *or later. For information about BGP configuration for products using AOS firmware prior to these releases, refer to Configuring BGP in AOS for Releases Prior to 18.03.00/R10.1.0, available online at* https://supportforums.adtran.com.

This configuration guide describes the configuration and use of Border Gateway Protocol (BGP) in AOS products. This guide includes an overview of BGP, the steps necessary to configure BGP, configuration examples for BGP, and additional troubleshooting information.

This guide consists of the following sections:

BGP Overview

BGP is an Exterior Gateway Protocol (EGP) that is used within the Internet and multinational organizations. EGP is one of two different types of dynamic routing protocols. The other protocol is Interior Gateway Protocol (IGP). The difference between the two protocols is that IGPs (for example, Routing Information Protocol (RIP), Open Shortest Path First (OSPF)) operate within an autonomous system (AS), whereas EGPs allow routes to be exchanged between different autonomous systems. Typically, an AS is defined by the boundaries of an organization. As an EGP, BGP routers must regulate traffic between networks controlled by organizations with different policies. BGP is designed to allow administrators to customize policies for route exchange. The following are some characteristics of BGP that make it an appropriate protocol for connecting different autonomous systems:

- BGP can filter both the routes it receives and those that it sends according to bit length, thereby minimizing the number of routes exchanged.
- BGP uses policies to determine best routes rather than per-hop counts used in RIP or link states used in OSPF. Each AS can set their own policy.
- BGP routers communicate only with manually configured neighbors.
- You can configure different policies for route exchange with different neighbors.
- BGP supports multiple virtual private network routing and forwarding (VRF) instances, which allows each VRF from the service provider its own BGP session within the router, thus extending the VRF from the service provider to the router.
- BGP supports multiple protocols, including Internet Protocol version 4 (IPv4) and IPv6.

BGP Advantages

Static routing, OSPF, and RIP are simple to configure, have low overhead, and are well suited for medium-to-small networks. However, BGP offers several advantages, particularly in more complex environments:

- Unlike routers using static routing, routers running BGP can automatically respond to connections that are down and changes in network topology. Multiple protocol label switching (MPLS) networks allow an organization to change its IP addressing scheme without notifying the service provider.
- BGP can handle complex applications in which the private network connects to multiple service provider routers or multiple service providers. BGP can be configured to balance loads among these connections.
- BGP is the native protocol implemented by service providers, which decreases problems caused by redistributing other routing protocols into BGP.
- BGP is policy based; therefore, organizations can maintain tight control over the routes transmitted and accepted.

Autonomous Systems

An AS is a group of networks administered by the same authority. Usually, an AS is the same as an organization. If the AS connects directly to the Internet, then the organization must acquire a unique number from the Internet Assigned Numbers Authority (IANA). However, many organizations connect to the Internet through a service provider who has already been assigned an AS number for connecting to the Internet. The service provider's AS is subdivided into areas and includes any organizations connecting to the Internet through them.

By defining autonomous systems, a demarcation point is created between organizations and the Internet. Within the AS, information about all networks can be transmitted to every other router on the network using some type of IGP, such as OSPF or RIP. However, since the Internet is so vast, it would be impractical for all routers to hold routes to all networks in all of the autonomous systems. The capability of the Internet to identify an entire organization by means of a unique integer allows a significant reduction in the amount of information that needs to be held in routing tables or transmitted in routing updates between autonomous systems. The result is a considerably smaller amount of summarized information exchanged using BGP between different autonomous systems. This level of hierarchy is essential to the successful operation and maintenance of the Internet.

Initially, AS numbers were only 16-bit integers. However, as the Internet continues to experience massive growth, engineers have expanded the AS number space from 2 bytes to 4 bytes, providing over 4 billion AS numbers. January 2009 marked the transition to allocate 32-bit AS numbers by default.

NOTE *As of release 18.1, AOS supports 32-bit AS numbers. As of release 18.03.00, AOS supports 4-byte AS numbers as part of the route distinguisher for VRFs. VRF routes in BGP can be distinguished by 2-byte AS number, 4-byte AS number, or by IPv4 address.*

External BGP (eBGP) and Internal BGP (iBGP)

eBGP uses BGP between peers in different autonomous systems, and iBGP uses BGP between peers within the same AS.

Do not confuse eBGP with EGP, a nearly obsolete protocol once used on the Internet. Also, do not confuse iBGP with an IGP, such as OSPF. Service providers use iBGP to distribute BGP routes between routers within an AS. However, service providers usually still need to run an IGP to generate routes for traffic within the AS.

eBGP allows an organization to peer and exchange routes with a service provider. MPLS implementations by a service provider allow the exchange of private subnets between remote sites through the provider's network. eBGP is also used between different service providers to facilitate the Internet backbone. Customers can peer using eBGP with service providers to exchange public IP addresses that they have purchased from IANA or to exchange private routes to other remote sites over an MPLS circuit.

iBGP is used between BGP routers within the same AS. iBGP routers prevent routing loops by following a rule where no updates learned from internal peers are sent to other internal peers. This means that iBGP routers will only propagate a route if the route originates in the transmitting router or if the route is directly connected to the transmitting router. As a result, iBGP routers must be fully meshed to have a complete knowledge of the network.

NOTE *AOS devices have a 16 BGP neighbor limit. Since iBGP routers must be fully meshed, AOS devices are only suitable for small scale iBGP applications.*

Route reflectors are sometimes identified and used in an iBGP network to reduce the number of peering sessions required.

NOTE *AOS devices cannot act as a route reflector for iBGP setups. AOS devices can peer with a route reflector acting as a client; however, the connection is no different than a regular iBGP peer.*

Address Families

Address families (AFs) are used in BGP to maintain a separation between Internet protocol types within a VRF. In 18.03.00, AOS implemented the use of AFs into the BGP hierarchical structure. An AF is a configuration structure that can reside at the default VRF and within a nondefault VRF. Creating an address family enables processing of that address family within a VRF, and it provides a place for AF-specific configuration.

As of AOS firmware release R10.1.0, two AFs can be specified (IPv4 and IPv6). When configuring BGP, you can configure global BGP settings for the router, and then create an AF-specific configuration for BGP behavior. Using the AF allows you to specify certain BGP behavior on a specific VRF.

BGP and VRFs

In addition to the use of AFs, multi-VRF BGP also allows the recognition and configuration of VRF specific to BGP functionality within the router. This allows the VRF constructs from the service provider to be incorporated into the customer router. To the service provider, there is little difference between connecting multiple customer edge (CE) routers (as in traditional BGP), and connecting to multiple VRFs on the CE using subinterfaces on a single CE router. Both the service provider and customer routers have a single BGP process and AS number, but they maintain a separate VRF-based session with neighbor addresses reachable within the scope of the individual VRF. Refer to *Multi-VRF BGP on page 8* for more information regarding the benefits and uses of VRFs in BGP.

BGP Neighbors and Messages

Unlike other routing protocols, BGP does not automatically discover its neighbors. The transport medium for BGP is Transmission Control Protocol (TCP), port 179. TCP is a connection-oriented protocol; thus, providing an extra layer of reliability for BGP communication.

BGP neighbors must be manually configured. After the three-way TCP handshake has been established between two BGP neighbors, a peering session is established through an open message that contains a hold time and BGP router ID. During the exchange of the open message, a router will decide if their neighbor is in the same AS (iBGP) or a different AS (eBGP). Once a session is established, keepalive messages are periodically exchanged between the peers to maintain connections and verify paths held by the router sending the keepalive. BGP neighbors can be configured globally, through an AF, or through a VRF.

BGP routers send update messages to their neighbors whenever a path changes. There is only one path per update. Each update message contains information on the path to a destination network and the path attributes, such as origin, AS path, and neighbor. Routes that are no longer available or withdrawn routes are included in updates.

Notification messages are sent when an error has been received. The BGP connection is closed after the notification message has been sent.

BGP Attributes

BGP attributes are properties that are used to determine the best route to a destination when multiple paths exist to a single destination. An understanding of how BGP attributes influence route selection is important when designing networks.

The following BGP attributes are supported by AOS:

- LOCAL_PREF
- MULTI_EXIT_DISC or MED
- Origin
- AS_PATH
- NEXT_HOP
- Community

Local Preference

The local preference (LOCAL_PREF) attribute is used to choose a path when there are multiple exit points from the local AS. Adjusting the local preference value can affect the local router’s decision, as well as the decision of other iBGP routers in the same AS when determining the best route to a destination.

> NOTE *The higher the local preference, the more desirable a route becomes.*

Multi-Exit Discriminator

The multi-exit discriminator (MULTI_EXIT_DISC or MED) attribute is used to influence eBGP neighbors to select a certain path for inbound traffic into the AS that is advertising the metric.

> NOTE *The lower the MED metric, the more desirable a route becomes.*

> NOTE *The MED metric is merely a suggestion to the external neighbor as to which path should be used inbound to the local AS. This is because the external AS that is receiving the MED metric may be using other BGP attributes for route selection.*

Origin

The origin attribute identifies the source of a learned route. There are three possible values for this attribute:

- **IGP** - The route has been learned by an iBGP neighbor that is internal to the local AS. This value is set when the **network** command is issued from BGP Configuration mode to inject a route into BGP.
- **EGP** - The route has been learned from an eBGP neighbor.
- **Incomplete** - The origin of the route is not known or learned in some other way. A route that has been redistributed into BGP or specified using the **network** command is set to this value.

AS Path

The AS path (AS_PATH) attribute consists of an ordered list of AS numbers that the route advertisement has crossed. Each time a router advertisement passes through an AS, the AS number is added to the list. The AS_PATH attribute is used by BGP as a loop avoidance mechanism. A route is rejected anytime a router detects its own AS number in a route advertisement.

The AS path can also be used to determine the best route to a given destination. If two identical routes are learned and all other attributes are equal, the one with the fewest number of traversed autonomous systems is preferred.

Next Hop

The next hop (NEXT_HOP) attribute is the IP address that is used to reach the advertising router. Since BGP routes traffic from AS to AS, the default next hop that is advertised is the address of the peer advertising the route from a remote AS.

Community

The community attribute provides a way to group routes together in communities and apply a consistent policy to the group. The policies can be set to control routing decisions, such as acceptance, preference, and redistribution.

> NOTE *For more detailed information on many of these BGP attributes and the related commands used to influence path selection, refer to Additional BGP Configuration on page 17.*

BGP Path Selection

When BGP receives advertisements for the same route from multiple sources, one path is selected as the best path and stored in the routing table. The decision logic used by BGP to determine the best path is fairly extensive. The following BGP criteria are used in AOS to select the best path to a destination:

1. Prefer the path with the higher LOCAL_PREF value.
2. If the LOCAL_PREF value is identical, compare the local-origination status. Prefer a route injected into BGP via the **network** <*ipv4 address*> **mask** <*subnet mask*> or **network** <*ipv6 address/prefix-length*> command issued from BGP AF configuration mode over a redistributed route.
3. If the local origination status is identical, prefer the shortest AS_PATH distance.
4. If the AS_PATH distance is identical, prefer the lower origin type (where routes originally injected via the **network** <*ipv4 address*> **mask** <*subnet mask*> or **network** <*ipv6 address/prefix-length*> command issued from BGP AF configuration mode or aggregation (IGP) are lower in origin than routes learned from a neighbor using eBGP. Routes originally injected by redistribution into BGP (incomplete) have the highest origin value).
5. If the origin type is identical, prefer the route with the lowest MULTI_EXIT_DISC value.
6. If the MULTI_EXIT_DISC value is identical, prefer eBGP paths over iBGP paths.
7. If the paths are still identical, prefer the path through the closest IGP neighbor.
8. Compare and prefer the lower value for any other metrics on the route.
9. Compare and prefer the route from the router with the lowest router ID.
10. Compare and prefer the route that came from the lowest neighbor IP address.

VRF and MPLS

The following information is provided to enhance the understanding of how service providers are able to maintain separation of private routes that belong to different customers.

A service provider uses virtual routing and forwarding (VRF) to separate one customer's routes from another's and MPLS to ensure that the routes reach only the authorized remote sites. Without VRF, customers could not transmit private network routes between remote sites; the service provider’s routers would have no way of knowing which route belonged to which customer.

For example, in *Figure 1 on page 7* the provider’s edge routers connect to two independent customers, Customer A and Customer B. Each customer would like to communicate private information between their own respective sites. Customer A Site 1 uses an IPv4 network address of 192.168.1.0 /24 and Customer A Site 2 uses an IPv4 network address of 192.168.2.0 /24. These identical network addresses are also used for Customer B Sites 1 and 2, respectively. The provider’s router must be able to associate one 192.168.1.0 /24 with the public address for Customer A Site 1 and the other with Customer B Site 1. Likewise, separate associations for 192.168.2.0 /24 for Customer A and B Site 2 must be maintained.

The provider’s edge router separates routes by the physical or logical interface on which they arrive. The router then stores routes from each customer in a separate VRF routing table. Different customers’ routing tables cannot be combined.

The service provider edge router connecting to the local site forms an MPLS label switched path (LSP) with the service provider edge router connecting to the authorized remote site. (An LSP resembles a dynamic permanent virtual circuit (PVC).) The edge routers mark packets with an MPLS label that directs them toward the other router through the LSP so that only Customer A sites receive Customer A routes.

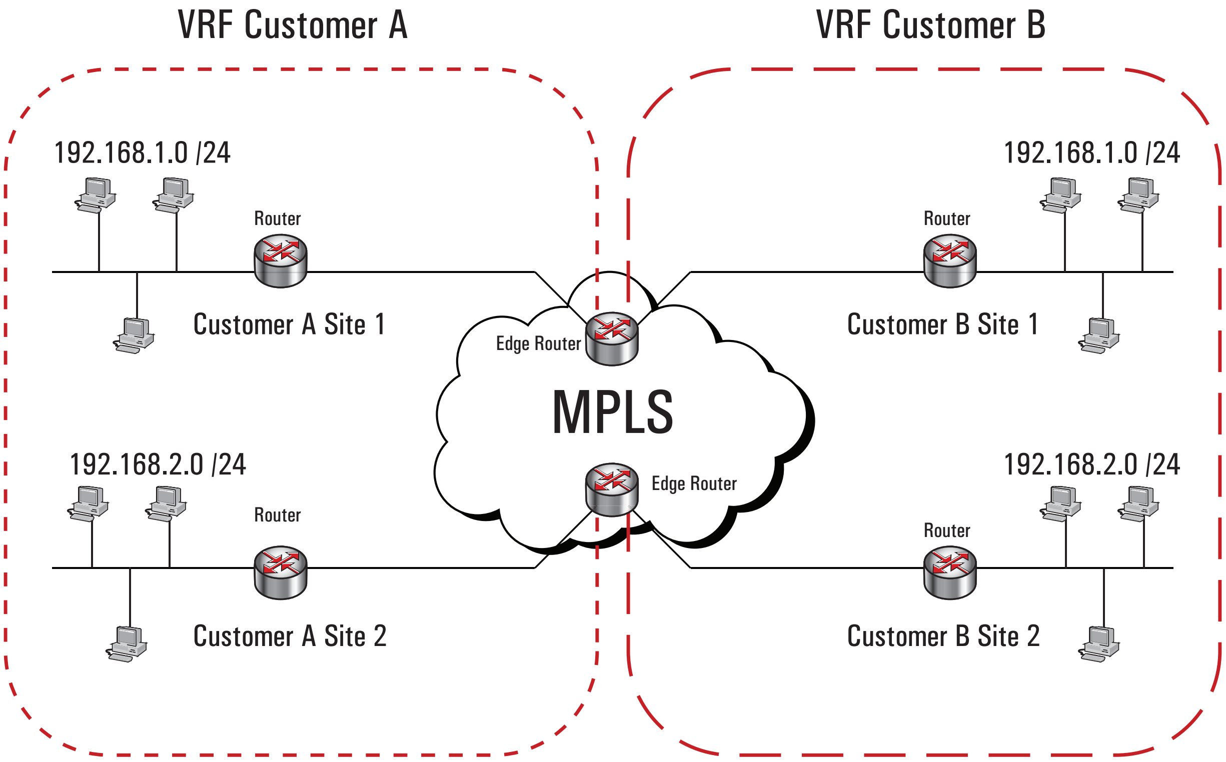


Figure 1. IPv4 Traffic Separation Using VRF

Multi-VRF BGP

The VRF functionality of BGP in service provider routers has been extended to the AOS router. The functionality of multi-VRF BGP is described in the following section.

Multi-VRF BGP is an application of the typical BGP functionality. As the name suggests, multi-VRF BGP extends traditional service provider multi-VRF functionality of BGP to the customer edge router. This type of feature is typically used in Layer 3 virtual private network (VPN) applications where the VPN is extended to the customer device using multi-VRF. Multi-VRF BGP allows the customer edge router to dynamically exchange customer VPN routes to and through the provider's VPN *cloud*, thus eliminating the reliance on the provider to manage static routes. While Layer 3 VPN specifications, such as BGP and MPLS, use Multi-Protocol BGP to convey MPLS labeled VPNv4 addresses on the core side of service provider routers, multi-VRF BGP takes place on the link between the service provider's routers and the customer edge equipment. Service providers' routers use this functionality to convey addresses between the customer's router and the customer's VRF on the service provider's router. This is the typical application using VRFs and BGP as described in *Figure 1 on page 7*.

When a set of VPNs is extended to the customer edge router, using multi-VRF, multi-VRF BGP can be implemented at the customer edge device. Using multi-VRF BGP allows each VRF its own BGP session with the respective VRF on the service provider's device. In this scenario, VPNs are separated on the provider-customer link by subinterfaces. To the service provider, there is little difference between connecting to multiple customer edge routers and connecting to multiple VRFs through subinterfaces on a single customer edge router. Both routers have a single BGP process and AS number, but maintain separate VRF-based sessions with neighbor addresses reachable within the scope of the individual VRFs. The function of multi-VRF BGP on the provider-customer link is described in *Figure 2 on page 8*.

In this scenario, the provider edge to customer edge (PE-CE) link has no MPLS, but rather VPN traffic is separated by subinterfaces (virtual local area network (VLAN), Frame Relay, Asynchronous Transfer Mode (ATM), etc.). In addition, in this type of configuration, the AOS router does not specify route targets (an extended community attribute).

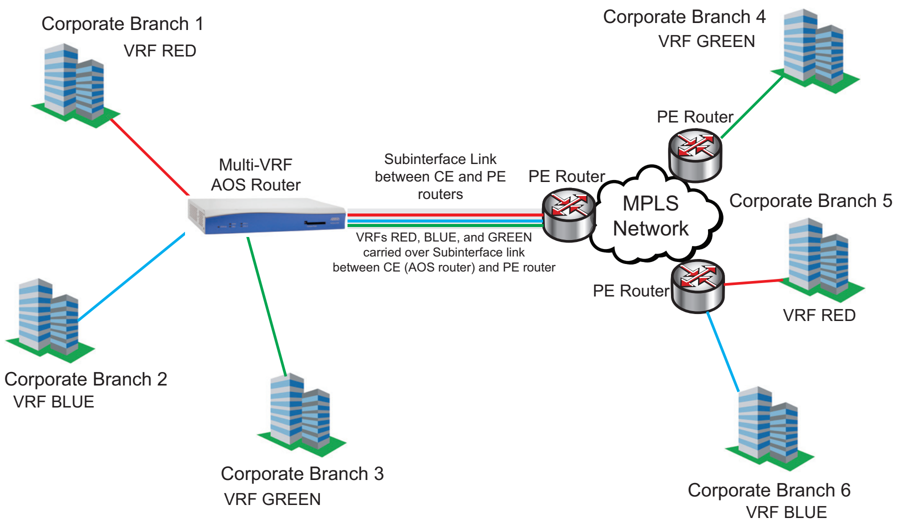


Figure 2. Multi-VRF BGP Network Topology

Multi-VRF BGP Terminology

The following are terms with which you should be familiar before using BGP.

- **Default VRF** - The global route table that exists even when no VRFs are defined. This implicit VRF has no assigned name, but when it is referenced it is identified as the **Default** VRF.
- **Network Layer Reachability Information (NLRI)** - The NLRI is typically an IP prefix advertised in a BGP UPDATE message.
- **Route** - The combination of a set of destinations as indicated by the IP prefix in the NLRI field of a BGP UPDATE message and the information reported in the path attributes field of the same UPDATE message.
- **Address Family Identifier (AFI)** - In combination with the Subsequent Address Family Identifier field, the AFI identifies the set of network layer protocols to which the address carried in the next hop packet field must belong; it identifies the way in which the address of the next hop is encoded, and it defines the semantics of the NLRI that follows.
- **Subsequent Address Family Identifier (SAFI)** - In combination with the AFI, this field identifies the set of network layer protocols to which the address carried in the next hop packet field must belong, the way in which the address of the next hop is encoded, and the semantics of the NLRI information that follows. If the next hop is allowed to be from more than one network layer protocol, then the encoding of the next hop MUST provide a way to determine its network layer protocol.
- **OPEN** - This message is sent from each BGP peer over the TCP session to begin BGP processing. BGP capabilities are negotiated in this exchange.
- **UPDATE** - This message is sent over TCP from a BGP peer to advertise or withdraw an address. The UPDATE message contains the address’s path attributes when being advertised.
- **NOTIFICATION** - This message is sent over TCP when an error condition is detected. The BGP connection is closed immediately after the message is sent.
- **KEEPALIVE** - These messages are exchanged (over TCP) between peers often enough that the hold timer will not expire.

BGP Organization

In BGPv4, commands are issued from the Global Configuration mode when configuring the global BGP parameters, from the BGP Configuration mode when configuring specific BGP behavior (such as AS number, local networks, and route information), and from the BGP Neighbor Configuration mode when configuring BGP neighbor parameters. With the incorporation of multi-VRF, multi-protocol, and IPv6 BGP, these basic hierarchical structures have changed.

In BGP configuration for AOS, the concepts of the AF and the VRF are incorporated into the BGP configuration structure. VRFs are in themselves not tied to a specific protocol, and therefore, control of multiple protocols within a single VRF is accomplished using an AF. BGP also uses the concepts of AFs to provide specific BGP configuration to a single BGP policy, and maintain control over multiple BGP policies within a single AF.

The hierarchical structure of BGP configuration is as follows:

```
Global BGP settings (affecting all of BGP for this router)
!
VRF BGP settings (affecting only the default VRF)
Neighbor #1 Definition
    Neighbor #1 common settings (common to all peering with neighbor #1)
    exit
!
Address Family for the default VRF
    AF BGP settings (affecting only this AF)
    Neighbor #1 Reference
        Neighbor #1 AF-specific settings
        exit
    exit
!
VRF name1
    VRF BGP settings (affecting only this VRF)
    Neighbor #2 Definition
        Neighbor #2 common settings (common to all peering with neighbor #2)
        exit
!
    Address Family for VRF NAME1
        AF BGP settings (affecting this AF)
        Neighbor #2 Reference
        Neighbor #2 settings specific to this AF
        exit
    exit
exit
```

In this type of configuration, you first enable BGP and configure the global BGP settings (including the local AS number (specified when BGP is enabled), description, source updating preferences, routing preferences, etc.). The router then assumes you are configuring settings for the default VRF, unless you configure a named VRF other than the default. If you are using the default VRF, you then configure the BGP settings for the VRF and specify a BGP neighbor and the neighbor settings. Unless the neighbor is configured in an AF, all neighbor settings are common to all devices peering with the neighbor. Once the global BGP, default VRF, and neighbor settings are configured, you begin configuring the AF for the default VRF. At this point you will configure the BGP AF settings, and the BGP neighbor settings specific to the AF. After configuring the AF for the default VRF, you can optionally configure additional VRFs for BGP operation. These additional VRFs are configured by name, by BGP settings, and by creating a neighbor for the specified VRF. The last step is to configure the AF settings for the VRF, and the BGP neighbor settings specific to the AF.

Hardware and Software Requirements and Limitations

eBGP is supported in AOS products running version 8.1 or later.

iBGP is supported in AOS products running version 10.1 or later.

Multi-VRF BGP is supported in AOS products running version 18.03.00 or later.

Multi-protocol and IPv6 BGP are supported in AOS products running version R10.1.0 or later.

eBGP, iBGP, and multi-VRF BGP are available on AOS data products as outlined in the *Product Feature Matrix* (ADTRAN's Knowledge Base article 2272).

> NOTE *AOS devices have a 16 BGP neighbor limit. Since iBGP routers must be fully meshed, AOS devices are only suitable for small scale iBGP applications.*

> NOTE *Support for the full Internet forwarding information base (FIB) from two peers requires either a NetVanta 4430 or NetVanta 5305 with 512 MB of random access memory (RAM).*

Basic BGP Configuration Using the CLI

There are several commands that must be issued for BGP to operate at the basic level, and several commands that enter into specific BGP configuration modes (AF, VRF, etc.). The following steps outline the minimum configuration required to enable BGP on an AOS device. Additional commands for each configuration mode are outlined in *Additional BGP Configuration on page 17*.

Step 1: Access the CLI

To access the CLI on your AOS unit, follow these steps:

1. Boot up the unit.
2. Telnet to the unit (**telnet** <*ip address*>), for example:

   ```
   telnet 10.10.10.1.
   ```

   > NOTE *If during the unit's setup process you have changed the default IP address (10.10.10.1), use the configured IP address.*

3. Enter your user name and password at the prompt.

   > NOTE *The AOS default user name is **admin** and the default password is **password**. If your product no longer has the default user name and password, contact your system administrator for the appropriate user name and password.*

4. Enable your unit by entering **enable** at the prompt as follows:

   ```
   >enable
   ```

5. If configured, enter your Enable mode password at the prompt.
6. Enter the unit's Global Configuration mode as follows:

   ```
   #configure terminal
   (config)#
   ```

Step 2: Enable BGP and Specify the Global BGP Settings

The first step in configuring BGP is to enable BGP and specify the local AS number. Use the **router bgp** command from the Global Configuration mode as follows:

(config)#**router bgp** *<AS number>*

> *<AS number>* Specifies the AS number of the local system of which this BGP router is a member. Range is **1** to **4294967295**.

Once you have entered this command, you are now in the BGP Configuration mode. From this mode, you can specify the global BGP settings for the AOS device. These settings include a hold timers, handling MEDs, router IDs, and more. Refer to *Additional BGP Global Settings on page 17* for more information.

Step 3: Configure the Default VRF BGP Settings

Once you have enabled BGP, specified the local AS number, and configured any global BGP settings, you can configure the default VRF BGP settings. Because you do not have to enter a specific VRF configuration mode for the default VRF, any BGP configurations made from the BGP Configuration mode (in Step 2), and any BGP neighbor configurations or AF configurations made from the BGP Configuration mode, are automatically applied to the default VRF.

Therefore, after configuring the global BGP configurations, you will need to create a BGP neighbor and an AF for the default VRF. BGP is different from many routing protocols because it does not allow a router to automatically search for peers from which to obtain routes. A separate BGP neighbor must be configured for each router with which the local router will communicate.

To create a BGP neighbor for the default VRF, enter the **neighbor [***<ipv4 address>* | *<ipv6 address>***]** command from the BGP Configuration mode to enter the BGP Neighbor Configuration mode. From this mode you can specify the neighbor's description, multihop capabilities, hold timers, the local AS, set a password, and many other options. At a minimum you must configure the neighbor's ID and remote AS number, and activate the neighbor.

A. Set the BGP IP address

BGP identifies a peer router by its IP address. Enter the BGP Neighbor Configuration mode from the BGP Configuration mode as follows:

(config-bgp)#**neighbor [***<ipv4 address>* **|** *<ipv6 address>***]**

> *<ipv4 address>* Specifies the IPv4 address for the neighbor. IPv4 addresses should be expressed in dotted decimal notation (for example, **10.10.10.1**).
>
> *<ipv6 address>* Specifies the IPv6 address for the neighbor. IPv6 addresses should be expressed in colon hexadecimal format (**X:X:X:X::X**), for example: **2001:DB8:1::1**.

NOTE *The IP address entered in this command must match the address for the interface that the remote router is using as its update source.*

The local router must be able to reach the IP address configured as the neighbor IP address. View the routing table and verify that it includes a route to this address.

B. Specify the Remote AS Number

The AS to which this neighbor belongs must also be specified. The **remote-as** command is issued from the BGP Neighbor Configuration mode:

```
(config-bgp-neighbor)#remote-as <value>
```

<value> Specifies the AS number. Range is **1** to **4294967295**.

When configuring eBGP, the remote AS number must be different from that of the local router (which is defined using the ***router bgp*** *command).*

The following example configures a remote AS number of **200** for IPv4 neighbor **172.16.1.2** on the default VRF:

```
(config)#router bgp 1
(config-bgp)#neighbor 172.16.1.2
(config-bgp-neighbor)#remote-as 200
```

You can add up to eight neighbors on the default VRF. Optional policies for each neighbor can be configured to dictate which routes the BGP interface sends to and accepts from the neighbor (refer to *Additional BGP Neighbor Settings (for both default VRF and nondefault VRF) on page 17*).

You can have up to eight neighbors on the default VRF, and up to eight neighbors on additional VRFs, but you can only have a total of 16 neighbors configured.

C. Activate the Neighbor

The BGP neighbor must be activated using the **no shutdown** command from the BGP Neighbor Configuration mode. To activate the neighbor on the default VRF, enter the command as follows:

```
(config-bgp-neighbor)#no shutdown
```

D. Configure the AF for the Default VRF

Once you have configured the BGP neighbor for the default VRF, you must configure the AF for the default VRF as well. To create an AF on the default VRF, enter the **address-family [ipv4 | ipv6]** command from the BGP Configuration mode prompt. The **ipv4** parameter specifies an IPv4 AF, and the **ipv6** parameter specifies an IPv6 AF. Using this command creates the AF and enters the AF's configuration mode. Enter the command as follows:

```
(config-bgp)#address-family ipv4
(config-bgp-ipv4)#
```

OR

```
(config-bgp)#address-family ipv6
(config-bgp-ipv6)#
```

Once you have entered the AF Configuration mode for the default VRF, you should advertise local networks that remote sites should be able to access. The following commands are used from the BGP AF Configuration mode to allow the BGP policy to advertise a network:

For IPv4 AFs:

```
(config-bgp-ipv4)#network <ipv4 address> mask <subnet mask>
```

> *<ipv4 address>* Specifies the IPv4 network address for the neighbor that AOS will advertise over BGP. IPv4 addresses should be expressed in dotted decimal notation (for example, **10.10.10.0**).
>
> *<subnet mask>* Specifies the subnet mask that corresponds to a range of IPv4 addresses (network) or a specific host. Subnet masks can be expressed in dotted decimal notation (for example, **255.255.255.0)**.

For example, to advertise the private network **10.1.10.0 255.255.255.0** for the IPv4 AF, enter:

```
(config-bgp-ipv4)#network 10.1.10.0 mask 255.255.255.0
```

> *The subnet mask is an integral part of the network address. If the BGP interface is specified to advertise routes to IPv4 network 10.1.0.0 /20, it will not advertise routes to network 10.1.0.0 /16 or 10.1.0.0 /24. Therefore, when advertising a network or range of networks, it must be verified that the routing table includes the exact route that has been specified (including the same subnet mask).*

For IPv6 AFs:

```
(config-bgp-ipv4)#network <ipv6 address/prefix-length>
```

> *<ipv6 address/prefix-length>* Specifies the IPv6 address and subnet for the neighbor that AOS will advertise over BGP. IPv6 address and prefixes are expressed in colon hexadecimal format (for example, **2001:DB8:0:3F3B::/64**).

For example, to advertise the private network **2001:DB8:0:3F3B::/64** for the IPv6 AF, enter:

```
(config-bgp-ipv6)#network 2001:DB8:0:3F3B::/64
```

BGP is a classless protocol. Therefore, networks with variable length subnet masks can be specified.

Since the BGP router is advertising a route, it searches its routing table for a route to the specified networks. It then sends this route to all authorized neighbors.

If the routing table does not include a route that has been specified in BGP, a null route must be configured. For example, a routing table only includes routes to the 24-bit networks, but not to the 20-bit network that contains them all. A route to IPv4 network 10.1.0.0 /20 must be manually added so that the BGP interface can advertise it. The route is added from the Global Configuration mode with the **null 0** keyword indicating the next-hop address:

```
(config)#ip route 10.1.0.0 255.255.240.0 null 0
```

> *If a more specific route (for example, from connected interfaces, static routes, routing protocols, etc.) other than the one to null 0 does not exist in the routing table, then all traffic for the specified subnet will be dropped. Null 0 routes should only be used when a more specific route is available to which to direct traffic.*

After you have specified which local addresses are to be advertised for the AF on the default VRF, you can specify multiple BGP attributes for the AF. These attributes include defining an administrative distance, specifying maximum allowed parallel paths, adding routes for advertisement, and redistributing routes between domains. These options are discussed in more detail in the section *Additional BGP AF Settings (for both default VRF and nondefault VRF) on page 18*.

E. Configure Default VRF AF Neighbor

You must specify a neighbor specific to the AF on the default VRF. You can have multiple neighbors specified for the AF, which will operate using the configurations specific to the AF to which they are assigned. To create a neighbor for the AF, enter the **neighbor [***<ipv4 address>* | *<ipv6 address>***]** command from the BGP AF Configuration mode as follows:

```
(config-bgp-ipv4)#neighbor 172.16.1.2
(config-bgp-ipv4-neighbor)#
```

OR

```
(config-bgp-ipv6)#neighbor 2001:DB8:1::1
(config-bgp-ipv6-neighbor)#
```

This command enters the BGP AF Neighbor Configuration mode. From this mode you can specify the neighbor's advertisement interval, apply filters, insert communities to outgoing BGP route updates, and configure per-neighbor soft reconfiguration. You can add multiple neighbors to an AF, but you should remember that each configuration is specific only to the neighbor for that AF. In addition, AF neighbors are not configured with a remote AS.

As with BGP neighbors on the default VRF, AF neighbors must also be activated using the **no shutdown** command from the AF neighbor's configuration mode. To activate the IPv4 AF neighbor on the default VRF, enter the command as follows:

```
(config-bgp-ipv4-neighbor)#no shutdown
```

Once the neighbor is defined in both the VRF and VRF AF, the route is advertised as described in *D. Configure the AF for the Default VRF on page 13*.

Additional AF neighbor configuration options are discussed in more detail in the section *Additional BGP AF Neighbor Settings (for both default VRF and nondefault VRF) on page 18*.

The IPv4 BGP configuration for a default VRF appears similarly to this example:

```
(config)#router bgp 1
(config-bgp)#neighbor 172.16.1.2
(config-bgp-neighbor)#remote-as 200
(config-bgp-neighbor)#no shutdown
!
(config-bgp)#address-family ipv4
(config-bgp-ipv4)#network 10.1.10.0 mask 255.255.255.0
(config-bgp-ipv4)#neighbor 172.16.1.2
(config-bgp-ipv4-neighbor)#no shutdown
```

Step 4: Configure the Nondefault VRF BGP Settings (Optional)

After configuring the global BGP settings, the default VRF settings, the default VRF neighbor, and the default VRF AF and AF neighbor, you can optionally configure additional VRF BGP settings. These configurations are completed using the same commands and steps as for the default VRF, however, the nondefault VRF must be created before configuring BGP policies, neighbors, or AF parameters. To create a nondefault VRF for use with BGP, enter the following command from the Global Configuration mode:

(config)#**vrf** *<name>* **route-distinguisher [as-2byte** *<ASN:nn>* **| as-4byte** *<ASN:nn>* **|**
ip *<ipv4 address:nn>***]**

vrf *<name>* Specifies the VRF name for the new VRF instance.

route-distinguisher Specifies the route distinguisher used by the VRF. All nondefault VRFs must have a route distinguisher specified. The route distinguisher is an 8-byte number that is prepended to an IPv4 address, allowing differentiation between overlapping addresses. The combination of the route distinguisher plus an IPv4 address is referred to as the VPNv4 address.

as-2byte *<ASN:nn>* or **as-4byte** *<ASN:nn>* Specifies the autonomous system number (ASN) relative route distinguisher as a 16-bit AS number and a 32-bit arbitrary number (4 byte), or that the ASN-relative route distinguisher is a 32-bit AS number and a 16-bit arbitrary number (2 byte). Route distinguishers are entered in the ASN:nn format, where ASN is either the 16-bit (2 byte) or 32-bit (4 byte) ASN, and nn is either a 32-bit arbitrary number (2 byte) or a 16-bit arbitrary number (4 byte).

ip *<ipv4 address:nn>* Specifies an IPv4 address relative route distinguisher, which consists of an IPv4 address and a 16-bit arbitrary number (*nn*, 2 byte number).

For example, to create a VRF **RED1**, with a route distinguisher based on IPv4 address **192.17.250.24** and arbitrary number **33**, enter the command as follows:

```
(config)#vrf RED1 route-distinguisher ip 192.17.250.24:33
```

For more specific information about both default and nondefault VRFs and their configuration and operation, refer to the configuration guide *Multi-VRF in AOS*, available online at https://supportforums.adtran.com.

Once you have configured the nondefault VRF, you can configure the BGP settings for the VRF using the **vrf** *<name>* command from the router's BGP Configuration mode. For example, to configure the BGP characteristics for the nondefault VRF **RED1**, enter the command as follows:

```
(config-bgp)#vrf RED1
(config-bgp-vrf)#
```

You have now entered the BGP VRF configuration mode for the named VRF. Once you are in this mode, you can configure the BGP settings for this specific VRF, create a neighbor for this VRF, and create an AF for this VRF, as well as an AF neighbor. These configurations follow the same steps as outlined in *Step 3: Configure the Default VRF BGP Settings on page 12*, except that they are performed for the named VRF instance. Additional VRF BGP configuration settings are outlined in *Additional BGP Global Settings on page 17*.

Step 5: Save the Configuration

The configuration can be saved directly from the BGP Configuration mode or BGP Neighbor Configuration mode:

```
(config-bgp-neighbor)#do write
```

NOTE *When the command to save the configuration is issued from the Enable mode, the command is:* ***write****.*

Additional BGP Configuration

Depending on the network, additional BGP configuration could be required. This section contains detailed explanations of additional BGP-related options that are available in AOS devices. Some configurations can be completed from multiple BGP configuration modes. For example, descriptions, passwords, hold timers, etc. can be configured from BGP or BGP Neighbor Configuration modes and applied globally, from BGP AF or BGP AF Neighbor Configuration modes, or from BGP VRF or BGP VRF Neighbor Configuration modes. The many additional BGP configuration options are outlined in the following sections.

Additional BGP Global Settings

These additional BGP configuration items are configured from the BGP Configuration mode. They apply globally to all BGP configurations in the AOS device, and by default apply to the default VRF BGP configuration. These configurations are accessed from the **(config-bgp)#** prompt.

- *Fast External Failover on page 18*
- *Hold Timer on page 19*
- *Multi-Exit Discriminators (MEDs) on page 19*
- *Log Neighbor Changes on page 19*
- *Router ID on page 20*

Additional BGP Neighbor Settings (for both default VRF and nondefault VRF)

These additional BGP configuration items are configured from the BGP Neighbor Configuration mode. They apply to all BGP neighbor configurations on the VRF and are common to all devices peering with the neighbor. These neighbor settings are configured on either the default VRF, from the prompt **(config-bgp-neighbor)#**, or on the nondefault (named) VRF, from the prompt **(config-bgp-vrf-neighbor)#**.

- *Description on page 20*
- *eBGP Multihop on page 20*
- *Hold Timer on page 19*
- *Local AS on page 21*
- *Password on page 21*
- *Transport Connection Mode on page 22*
- *Update Source on page 22*

Additional BGP AF Settings (for both default VRF and nondefault VRF)

These additional BGP configuration items are configured from the BGP AF Configuration mode. They apply to the AF for either the default VRF, with the prompt **(config-bgp-ipv4)#** or **(config-bgp-ipv6)#**, or for the nondefault (named) VRF, with the prompt **(config-bgp-vrf-ipv4)#** or **(config-bgp-vrf-ipv6)#**. These settings affect only the AF for which they are configured.

- *Distance on page 23*
- *Establishing Routing Preference on page 24*
- *Maximum Paths on page 27*

Additional BGP AF Neighbor Settings (for both default VRF and nondefault VRF)

These additional BGP configuration items are configured from the AF Neighbor Configuration mode. They apply to the neighbor configured on a specific AF.

AF neighbors can be configured from either the default VRF, with the prompt **(config-bgp-ipv4-neighbor)#** or **(config-bgp-ipv6-neighbor)#**, or from the nondefault (named) VRF, with the prompt **(config-bgp-vrf-ipv4-neighbor)#** or **(config-bgp-vrf-ipv6-neighbor)#**.

- *Advertisement Interval on page 27*
- *AS-Path List on page 28*
- *BGP Communities on page 29*
- *Distribute List on page 33*
- *Next-Hop Self on page 34*
- *No Default Originate on page 35*
- *Prefix List on page 35*
- *Route Maps on page 37*
- *Applying a Route Map Entry to a BGP Neighbor on page 46*
- *Soft Reconfiguration Inbound on page 46*

Additional Global BGP Settings

The following commands and configurations are used to configure the global BGP settings for the AOS device. These settings are applied across all BGP functionality for the device.

Fast External Failover

Use the **bgp fast-external-failover** command to enable the fast external failover feature.

```
(config-bgp)#bgp fast-external-failover
```

When failover is enabled, if the link interface goes down between this router and a BGP neighbor, the BGP session with the neighbor is immediately cleared. When failover is disabled and the link goes down, the session is maintained until the BGP hold timer expires (refer to *Hold Timer on page 19*).

Hold Timer

Use the **hold-timer** command to set the default hold time for BGP neighbors. The command can be issued in BGP Configuration mode to set the default hold time for all neighbors in that BGP process, or the command can be issued in BGP Neighbor Configuration mode to set the hold time for only that neighbor.

(config-bgp)#**hold-timer** <*value*>

or

(config-bgp-neighbor)#**hold-timer** <*value*>

> <*value*> Specifies a time interval (in seconds) within which a keepalive must be received from a peer before it is declared a dead peer. Range is **0** to **65535** seconds.

The peers will negotiate and use the lowest configured setting. The keepalive interval will be set to one-third of the negotiated hold time.

NOTE *The default hold time is **180** seconds.*

Multi-Exit Discriminators (MEDs)

Use the **bgp** command to instruct AOS on how to handle multi-exit discriminators (MEDs) for all routes from the same AS.

(config-bgp)#**bgp [always-compare-med | compare-med | deterministic-med | ignore-med]**

> **always-compare-med** Always compares MEDs for all paths for a route, regardless of the AS through which the route passes.
>
> **compare-med** Compares MEDs for all received routes.
>
> **deterministic-med** Compares the MEDs for all routes received from different neighbors within the same AS.
>
> **ignore-med** Disregards MEDs for all received routes.

NOTE *Refer to B. Setting a MED Metric on page 44 for information on how to configure the value of the MED metric advertised outbound to BGP neighbors. For general information about MED functionality, refer to BGP Path Selection on page 6.*

Log Neighbor Changes

Use the **bgp log-neighbor-changes** command to control the logging of neighbor state changes. This command controls logging of BGP neighbor state changes (up/down) and resets. This information is useful for troubleshooting and determining network stability.

(config-bgp)#**bgp log-neighbor-changes**

Router ID

The BGP interface identifies itself to neighbors with its router ID. Router IDs are configured on a per-VRF basis. There is only one router ID per VRF instance. This ID can be the IP address of the logical interface that connects to each neighbor, except when there are multiple interfaces with neighbors configured. The ID can also be the address of a loopback interface used as the update source. A loopback interface that is the update source for BGP ensures that a BGP session stays open even if one connection goes down. The following command specifies the router ID:

```
(config-bgp)#bgp router-id <ipv4 address>
```

<ipv4 address> Designates the IPv4 address this router should use as its BGP router ID. IPv4 addresses should be expressed in dotted decimal notation (for example, **10.10.10.1**).

> NOTE *If no IP address is configured at BGP startup, it uses the highest IP address configured on a loopback interface. If no loopback interfaces are configured, it uses the highest IP address configured on any interface that is active. If the specified router ID is changed, existing sessions with BGP neighbors are reset.*

Additional BGP Neighbor Settings

These additional BGP neighbor settings are configured from the BGP Neighbor Configuration mode, for either the default VRF or a nondefault (named) VRF. In addition to the settings listed below, hold timers can be configured for a specific neighbor, as described in *Hold Timer on page 19*.

Description

Use the **description** command to identify the specified interface connected to a BGP neighbor. This description is not used within the BGP protocol itself, but rather as a block of descriptive text seen in the BGP configuration. For a neighbor on a default VRF, enter the command as follows:

```
(config-bgp-neighbor)#description <text>
```

<text> Identifies the specified interface using up to **80** alphanumeric characters.

For a neighbor on a nondefault (named) VRF, enter the command as follows:

```
(config-bgp-vrf-neighbor)#description <text>
```

eBGP Multihop

Use the **ebgp-multihop** command to configure the maximum hop count for BGP messages to a neighbor. To configure eBGP multihop for a neighbor on the default VRF, enter the command as follows:

```
(config-bgp-neighbor)#ebgp-multihop <value>
```

<value> Specifies the maximum hop count of BGP messages to a neighbor. Range is **1** to **255** hops.

To configure eBGP multihop for a neighbor on a nondefault (named) VRF, enter the command as follows:

```
(config-bgp-vrf-neighbor)#ebgp-multihop <value>
```

This command allows a BGP neighbor to be on a network that is not directly connected. The default time to live (TTL) for BGP messages is **1** since eBGP peers are normally directly connected. However, in certain applications, a non-BGP device, such as a firewall or router, may reside between eBGP peers. The **ebgp multihop** command is required in this case to allow neighbor relationships to be formed when the neighbor is not directly connected.

It is also good practice to create a static route to reach the eBGP neighbor when the neighbor is more than one hop away, as well as creating a backup route pointing to the null 0 interface with a higher administrative distance. These routes prevent unnecessary BGP traffic from traversing the wrong connection and prevent the BGP neighbor communication from incorrectly attempting to establish itself on that connection. It is also possible that sending invalid BGP traffic to some providers may result in the connection being automatically disabled by the provider as a security measure.

Local AS

Some multihoming network designs require a customer to appear as a different AS number to individual service providers. Also, service providers sometimes assign the same AS to multiple sites, which can cause problems due to BGP’s loop avoidance check mechanism (refer to *Example 2 on page 49*). The **local-as** command rectifies both situations by substituting an AS number that is different from the one specified in the command **router bgp** <*AS number*>.

Use the **local-as** command to specify an AS number for the unit to use when communicating with this BGP neighbor on the default VRF.

(config-bgp-neighbor)#**local-as** <*value*>

> <*value*> Specifies the AS number to use when communicating with this neighbor. The value must be different than the AS number for this router and the peer router. It is only valid for eBGP connections. Range is **0** to **4294967295**. When **0** is used, it indicates that the BGP process local AS is used, because **0** is not a valid AS number.

> NOTE *By default, the **local-as** value is set to **0**, which means the BGP process’s AS number is used.*

To specify an AS number for the unit to use when communicating with this BGP neighbor on a nondefault (named) VRF, enter the command as follows:

(config-bgp-vrf-neighbor)#**local-as** <*value*>

Password

Use the **password** command to enable message digest 5 (MD5) password authentication on TCP segments exchanged with the BGP peer. Enter the command as follows for a BGP neighbor on the default VRF:

(config-bgp-neighbor)#**password** <*password*>

> <*password*> Specifies the password string to be used for authentication. The password is case sensitive and must not exceed **80** characters.

> NOTE *Authentication must be configured on both peers using the same password.*

Enter the command as follows for a BGP neighbor on a nondefault (named) VRF:

```
(config-bgp-vrf-neighbor)#password <password>
```

Every BGP TCP segment sent is authenticated. Configuring authentication causes an existing session to be torn down and re-established using the currently specified authentication.

Transport Connection Mode

Use the **transport connection-mode [active | passive]** command to specify BGP transport session options for the BGP neighbor. Active connections are those that are initiated by the router to establish a TCP connection to the neighbor. Passive connections are those in which the router will only listen for incoming BGP connections without trying to establish a connection. By default, both active and passive connections are supported simultaneously, and a collision detection algorithm is used to determine which TCP session to use should an inbound and outbound connection begin. Enter the command as follows for a BGP neighbor on the default VRF:

```
(config-bgp-neighbor)#transport connection-mode [active | passive]
```

active Specifies that only active TCP session connections are allowed for the neighbor.

passive Specifies that only passive TCP session connections are allowed for the neighbor.

To specify which connections are allowed for a BGP neighbor on the nondefault (named) VRF, enter the command as follows:

```
(config-bgp-vrf-neighbor)#transport connection-mode [active | passive]
```

Update Source

Use the **update-source** command to specify which interface's IP address will be used as the source IP address for the BGP TCP connection. To define the interface on the default VRF BGP Neighbor, enter the command as follows:

```
(config-bgp-neighbor)#update-source <interface>
```

<interface> Specifies the interface to be used as the source IP address. Specify an interface in the format *<interface type [slot/port | slot/port.subinterface id | interface id | interface id.subinterface id>*, for example, for an Ethernet interface, use **eth 0/1**; for a PPP interface, use **ppp 1**; and for an ATM subinterface, use **atm 1.1**. Enter **update-source ?** for a complete list of valid interfaces.

> NOTE *This command is most often configured as a loopback interface that is reachable by the peer router. The peer will specify this address in its neighbor commands for this router.*

To define the interface on a nondefault (named) VRF BGP neighbor, enter the command as follows:

```
(config-bgp-vrf-neighbor)#update-source <interface>
```

When an AOS device sends a BGP packet, the routing table is first consulted to determine how to reach the intended peer. By default, the IP address assigned to the egress interface of the local router the packet uses to reach the peer is the source address for the BGP packet. This address can be overwritten by using the **update source** command.

Loopback interfaces should be used when there are multiple paths to reach a single iBGP neighbor. Using a loopback address as the update source forces the BGP messages to use only the IP address of the loopback interface instead of the IP addresses associated with the egress interface of any of the multiple links from a single neighbor. The loopback interface is first created and then advertised as the update source toward the intended iBGP neighbor.

NOTE *Using a loopback interface as the update source for an iBGP neighbor is often done in conjunction with an interior routing protocol, such as OSPF, that can dynamically advertise routes to reach a particular destination when failures and topology changes occur.*

Loopback interfaces can also be used in eBGP scenarios where multiple links to the same neighbor exist and a multilink protocol is not being used. Similar to the iBGP scenario above, the loopback address would need to be configured and then specified with the **update-source** command on a per-neighbor basis. However, some additional configuration is required for eBGP applications. eBGP assumes that the neighbor is one hop away on a directly connected interface. When loopback addresses are used, an extra hop is created between the neighbors, thus requiring the use of the **ebgp-multihop** command to increase the TTL value for the IP packets carrying the BGP messages. If the value for this command is not changed in accordance to the new number of hops, the TTL for IP packets carrying the BGP messages intended for eBGP neighbors defaults to 1 and will expire before the packet reaches the destined loopback interface on the peer router. Refer to *eBGP Multihop on page 20* for more information on this command.

NOTE *When a loopback interface is used as the update source address, one extra hop is created between eBGP neighbors. It is important to account for this extra hop when calculating the TTL value set in the* ***ebgp-multihop*** *command.*

The peering router would also use a loopback interface in this scenario. The neighbor address configured on each router would be the IP address of the loopback interface on the peering router. Since the loopback address enables the BGP neighbors to use an IP address that is not reachable by a directly connected interface, separate static routes would need to be specified for each individual link that can reach the neighbor's loopback IP address.This ensures that the loopback address on the BGP peer remains reachable if one of the links goes down.

Additional BGP AF Settings

These additional BGP AF settings are configured from the BGP AF Configuration mode. The AF can be configured for either the default VRF, or a nondefault (named) VRF. The BGP settings for the AF are specific to that AF only.

Distance

The administrative distance is a local variable that allows a router to choose the best route when there are multiple paths to the same network. Use the **distance bgp** command to set the administrative distance for BGP routes. To set the administrative distance for an IPv4 AF on the default VRF, enter the command as follows:

(config-bgp-ipv4)#**distance bgp** *<external> <internal> <local>*

<external> Sets the administrative distance for BGP routes learned via eBGP sessions. Range is **1** to **255**.

<internal> Sets the administrative distance for BGP routes learned via iBGP sessions. Range is **1** to **255**.

<local> Sets the administrative distance for BGP routes learned via the network and redistribution commands. Range is **1** to **255**.

> NOTE *By default, external is set to **20**, internal to **200**, and local to **200**. Normally, these default settings should not be changed.*

> NOTE *Routes with lower administrative distances are favored.*

To set the administrative distance for an IPv4 AF on a nondefault (named) VRF, enter the command as follows:

(config-bgp-vrf-ipv4)#**distance bgp** *<external> <internal> <local>*

Establishing Routing Preference

When a route is redistributed into BGP from a static route, another routing protocol, or when a prefix is specifically advertised using a network statement in BGP Configuration mode, BGP must consider it in its route calculations.

Assuming the LOCAL_PREF value of the two routes is equal, the BGP decision logic prefers routes learned by the local router from statically configured routes or routing protocols over routes learned from a BGP peer.

> NOTE *Assuming both routes have an equal LOCAL_PREF value, it doesn't matter if the administrative distance of a locally learned route is configured higher than that of BGP. The local route will still be used instead of the route learned from the BGP peer. This scenario occurs only when the same route is learned locally and by BGP.*

After the best valid route has been selected by the BGP decision logic, the administrative distance for BGP is applied to the route. The routing table then chooses the best route to install based on the lowest administrative distance. If the BGP algorithm chooses a locally learned route with a higher administrative distance that is already installed in the routing table, it will not replace that existing route since it points to the same next-hop IP address. As this specific scenario is manifested, it can be observed in the output of the **show ip route** or **show ipv6 route** commands. The routing source will show the locally learned route from its original source instead of a BGP sourced route (denoted with a B).

> NOTE *Despite administrative distance and assuming equal LOCAL_PREF value, BGP learned routes are always less preferable than routes learned by the local router from statically configured routes or routing protocols.*

The best way to make a static route or a prefix learned from another routing protocol more or less preferable than any route learned using BGP is to redistribute the protocol into BGP (using the **redistribute [connected | ospf | rip | static] [metric** <*value*> **| route-map** <*map*>**]** command) and use a route map to modify the local preference of the intended routes. If this approach is used, an additional prefix list might need to be configured and applied outbound to all neighbors to prevent unwanted redistributed subnets from being advertised.

> NOTE *The* ***redistribute*** *command does not include the* ***ospf*** *and* ***rip*** *options in the nondefault (named) VRF.*

The default LOCAL_PREF value for BGP routes is **100**. To prefer the redistributed route over a route learned through BGP, the LOCAL_PREF of the redistributed route must be set to a value higher than 100. To prefer a route learned through BGP over a redistributed route, an inbound route map should be applied to the appropriate BGP neighbor to set the LOCAL_PREF of the intended route higher than 100.

The following example configuration demonstrates the basic commands used to prefer an IPv4 static route or a route learned through another routing protocol (local prefix) over a BGP learned route in the default VRF IPv4 AF:

```
!
route-map REDISTRIBUTE permit 10
    match ip address prefix-list <name>
    set local preference 110
!
router bgp <AS number>
    address-family ipv4
        redistribute <source protocol> route-map REDISTRIBUTE
        exit
    exit
!
```

The following example configuration demonstrates the basic commands used to prefer an IPv6 static route or a route learned through another routing protocol (local prefix) over a BGP learned route in the default VRF IPv6 AF:

```
!
route-map REDISTRIBUTE permit 10
    match ipv6 address prefix-list <name>
    set local preference 110
!
router bgp <AS number>
    address-family ipv6
        redistribute <source-protocol> route-map REDISTRIBUTE
        exit
    exit

!
```

*The **redistribute** command supports connected, OSPF, RIP, and static routes on the default VRF. These are all considered locally originated routes to BGP. However, locally learned routes are only considered in the BGP path selection process if the locally learned routes are active in the route table. For example, if a duplicate route is learned using OSPF and BGP, and the OSPF route is not active in the route table (which can occur if the OSPF route is learned after the duplicate route has first been learned by BGP), the OSPF route is not selected for the BGP path. This behavior can lead to different routes being in the route table depending on dynamic events such as the timing of routing protocol convergence and the losing and relearning of routes.*

The following example configuration demonstrates the basic commands used to prefer the IPv4 prefix learned through BGP over an IPv4 static route or a route learned through another routing protocol (local prefix) in the default VRF IPv4 AF:

```
!
route-map PREFER permit 10
    match ip address prefix-list <name>
    set local-preference 110
!
router bgp <AS number>
    address-family ipv4
        route-map PREFER in
        exit
    exit
exit
```

The following example configuration demonstrates the basic commands used to prefer the IPv6 prefix learned through BGP over an IPv6 static route or a route learned through another routing protocol (local prefix) in the default VRF IPv6 AF:

```
!
route-map PREFER permit 10
    match ipv6 address prefix-list <name>
    set local-preference 110
!
router bgp <AS number>
    address-family ipv6
        route-map PREFER in
        exit
    exit
exit
```

Example 7 on page 61 further demonstrates the use of the above example configurations to establish preferences for certain routes.

Maximum Paths

Use the **maximum-paths** command to specify the number of equal cost parallel routes (shared paths) learned by BGP that can be exported to the route table. When IP load sharing is enabled, traffic is balanced to a specific destination across up to six equal paths. To specify the number of shared IPv4 paths exported on the default VRF AF, enter the command as follows:

(config-bgp-ipv4)#**maximum-paths** <*value*>

> <*value*> Specifies the number of equal cost parallel routes learned by BGP that can be exported to the route table. Valid range is **1** to **6**. Default value is **1**. For proper configuration, set this value to **2** or more.

To specify the number of shared paths exported on a nondefault (named) VRF AF, enter the command as follows:

(config-bgp-vrf-ipv4)#**maximum-paths** <*value*>

To specify the number of shared routes for IPv6 routing on the default VRF AF, enter the command as follows:

(config-bgp-ipv6)#**maximum-paths** <*value*>

> <*value*> Specifies the number of equal cost parallel routes learned by BGP that can be exported to the route table. Valid range is **1** to **6**. Default value is **1**. For proper configuration, set this value to **2** or more.

Additional BGP AF Neighbor Settings

These additional BGP AF neighbor settings are configured in the BGP AF Neighbor Configuration mode, and are applied to either the default VRF AF neighbor, or the AF neighbor on a nondefault (named) VRF. These settings are applicable only to the specific AF neighbor.

Advertisement Interval

Use the **advertisement-interval** command to configure AOS to specify how long the BGP process waits before sending updates to the neighbor. This command sets the minimum interval between sending updates to the specified neighbor. To configure the advertisement interval for an IPv4 AF neighbor on the default VRF, enter the command as follows:

(config-bgp-ipv4-neighbor)#**advertisement-interval** <*value*>

> <*value*> Specifies the advertisement interval in seconds. Range is **0** to **600** seconds.

To configure the advertisement interval for an IPv4 AF neighbor on a nondefault (named) VRF, enter the command as follows:

(config-bgp-vrf-ipv4-neighbor)#**advertisement-interval** <*value*>

> NOTE *By default, the advertisement interval is* ***30*** *seconds for external neighbors and* ***5*** *seconds for internal neighbors.*

AS-Path List

IPv4 routes can be filtered according to the hops listed in the AS field. For advertised IPv4 routes, this type of filtering allows a degree of influence over which autonomous systems external neighbors can access. For example, service provider routers can filter routes with paths that include customer AS numbers to prevent themselves from advertising private customer routes to unauthorized peers. Private networks do not typically transmit traffic from AS to AS. Therefore, filtering advertised routes according to AS path is not usually necessary when configuring eBGP in a private network.

To filter routes for the AF neighbor based on the AS_PATH field, you can either specify the AS path list in the AF neighbor configuration, or create an AS-path filter in a route map (detailed in *Route Maps on page 37*). To apply a filter to the AF neighbor for either the default VRF or a nondefault (named) VRF without using a route map, follow these steps:

Step 1: Creating an AS-Path List

Create a list of the AS paths to be filtered using the **as-path-list** command from the Global Configuration mode prompt. Enter the command as follows:

```
(config)#as-path-list <name>
```

<name> Specifies the name of the AS path list.

Step 2: Specifying AS Numbers for Filtering and Permitting/Denying Routes

Next, specify the AS numbers to be filtered and specify whether the routes containing these AS numbers should be permitted or denied. Enter the command as follows:

```
(config-as-path-list)#[deny | permit] <value>
```

<value> Specifies permitting or denying routes that contain this value in their AS_PATH attribute. This is a string that follows the format of AS regular expressions to filter an AS path. Refer to *AS Regular Expressions on page 110* for a detailed list of valid AS regular expressions.

The AS path list is only compared against the AS_PATH attribute on a BGP prefix, which is also displayed in the output of the **show bgp** command.

For example, a router can be permitted to advertise only IPv4 routes that use both AS 200 and AS 400:

```
(config-as-path-list)#permit 200 400
```

NOTE *Using **permit 200 400** will match traffic destined to AS 200 and 400, but it could also match any number of autonomous systems that have 200 and 400 in them (for example, 2200 and 4000). To match only AS paths that include 200 and 400, use regular expressions to determine the match. For example, to match only AS paths that include 200 and 400 in any order, enter **permit (\b200\b.*400\b)|(\b400.b*200\b)**.*

However, the statement above only permits routes that use both AS 200 and AS 400. Permit any routes that use either AS by entering separate statements:

```
(config-as-path-list)#permit \b200\b
(config-as-path-list)#permit \b400\b
```

Permitting AS number 200 selects any IPv4 routes that include that value, even if the AS field also includes other values. In other words, entering **permit \b200\b** permits routes containing AS 200, as well as AS 200 and AS 400, while entering **permit (\b200\b.*400\b)|(\b400.b*200\b)** only permits routes containing both AS 200 and AS 400. Therefore, it could be necessary to explicitly deny any values that should not be included in the field.

Another example might be where the router is allowed to advertise IPv4 routes that use AS 200 or AS 400, but not routes that force traffic to travel through both AS 200 and AS 400:

```
(config-as-path-list)#deny (\b200\b.*400\b)|(\b400.b*200\b)
(config-as-path-list)#permit \b200\b
(config-as-path-list)#permit \b400\b
```

Step 3: Applying AS-Path List to a BGP AF Neighbor

After creating the IPv4 AS path list, you must apply it to the IPv4 AF neighbor. To apply the AS path list filter to the IPv4 AF neighbor on the default VRF, enter the **as-path-list** command from the BGP AF Neighbor Configuration mode as follows:

```
(config-bgp-ipv4-neighbor)#as-path-list <name> [in | out]
```

> *<name>* Specifies the name of the AS path list that will be used to filter routes on the AF neighbor. This is the AS path list created in Steps 1 and 2.
>
> The optional **in** keyword specifies the filter is applied to inbound data, the optional **out** keyword specifies that the filter is applied to outbound data.

To apply the IPv4 AS path list filter to the IPv4 AF neighbor on a nondefault (named) VRF, enter the command as follows:

```
(config-bgp-vrf-ipv4-neighbor)#as-path-list <name> [in | out]
```

BGP Communities

Within BGP, the community is an optional attribute that can be used for identification, security, or to signal a BGP peer that it should take a particular action. When used for identification and security, the attribute adds another layer of complexity that requires special configuration. When used to signal a peer, the attribute is commonly used when that peer is ignoring other attributes of the BGP advertisement (which is often the case in MPLS networks).

A route can be a member of one or more BGP communities. A community is simply a way of grouping routes together and applying a consistent policy to the group. A route can be placed into a community according to any attribute in that route. One of the most common ways of grouping routes is by network address and prefix length, which is defined in a prefix list and ultimately referenced in a route map (refer to *Prefix List on page 35* and *A. Filtering Routes According to Network Address on page 40*). In order for a route's membership in a community to have significance, administrators must define policies that apply to the community.

Several commands must be issued when configuring BGP communities on an AOS device. A route map must be configured, followed by the appropriate commands for sending and/or receiving a community string. Also, the **send-community standard** command must be enabled for any AF neighbor that will be sending or receiving community attributes (refer to *Step 3: Enabling an AOS Device to Send or Receive BGP Communities on page 32*).

Step 1: Configuring a Community List

A community list is used when an AOS device receives a community attribute in a BGP UPDATE message. A community list can be used to:

- Select the communities to which the router will apply a specific policy, such as filtering advertised routes to those communities or applying policies to inbound routes from those communities.
- Define the communities that the BGP process will delete from routes.

Use the **community-list** command to create a community list for BGP route map use. The communities defined in this command should match an existing community string that the AOS device receives. This command is issued from the Global Configuration mode:

(config)#**community-list** <*name*>

<*name*> Specifies the community list name. This is an arbitrary name for the list that is referenced in a BGP policy using community strings.

This command places the user in the Community List Configuration mode where one or more well-defined communities can be specified. A value for a privately defined community can also be specified.

The following command adds an entry to the community list that either **permits** or **denies** BGP routes containing the specified community string in the community attribute:

(config-comm-list)#**[permit | deny] [**<*value*> **| internet | local-as | no-advertise | no-export]**

<*value*> Specifies a privately defined community for routes that contain this value in their community attribute. This is a numeric value that can be an integer from **1** to **4294967295** or string in the form ***aa:nn***, where ***aa*** is the AS number and ***nn*** is the community number. Multiple community number parameters can be present in the command.

internet Specifies routes that contain the reserved community number for the Internet community.

local-as Specifies routes that contain the reserved community number for NO_EXPORT_SUBCONFED. Routes containing this attribute should not be advertised to external BGP peers.

no-advertise Specifies routes that contain the reserved community number for NO_ADVERTISE. Routes containing this attribute should not be advertised to any BGP peer.

no-export Specifies routes that contain the reserved community number for NO_EXPORT. Routes containing this attribute should not be advertised to BGP peers outside a confederation boundary.

Table 1 below summarizes the well-known communities and the policy expected for these communities.

Table 1. Well-Known Communities

Community	Advertise To
internet	All peers
local-as	Peers in the local AS
no-advertise	No peers
no-export	Internal peers only

> *Multiple communities can be specified by stringing several keywords in the same command (for example, (config-comm-list)#**permit local-as no-export**).*

After the community list has been defined, it must be referenced in a route map entry. Refer to *Route Maps on page 37* for information on creating route maps. Refer to *C. Filtering Routes According to Community on page 42* for information on how to reference a community list within the route map.

Step 2: Defining a Community Policy

The community policy defines the action that will be taken on routes depending on the community string attached to those routes. This can be used to set certain attributes, such as local preference or metric, for routes in the community. The **set** commands are used to define the community policy on a community string. These commands are issued within Route Map Configuration mode (refer to *Route Maps on page 37*) and the route map is applied outbound to a BGP AF neighbor (refer to *Applying a Route Map Entry to a BGP Neighbor on page 46*).

> *Before placing a route in a community, you should contact your service provider and discuss what options it supports for various communities. You should also consult your organization's policies.*

Use the **set community** command to select the peers to which a neighbor advertises routes in the community:

(config-route-map)#**set community** <*value*> **[add | internet | local-as | no-advertise | no-export]**

<*value*> Specifies a privately defined community for routes serviced by this route map. This is a numeric value that can be an integer from **1** to **4294967295** or string in the form ***aa:nn***, where ***aa*** is the AS number and ***nn*** is the community number. Multiple community number parameters can be present in the command.

add Appends the listed community number to the end of the community attribute for routes serviced by this route map.

internet Sets the community attribute to the reserved INTERNET community number for routes serviced by this route map.

local-as Sets the community attribute to the reserved NO_EXPORT_SUBCONFED community number for routes serviced by this route map. Routes containing this attribute should not be advertised to eBGP peers.

no-advertise Sets the community attribute to the reserved NO_ADVERTISE community number for routes serviced by this route map. Routes containing this attribute should not be advertised to any BGP peer.

no-export Sets the community attribute to the reserved NO_EXPORT community number for routes serviced by this route map. Routes containing this attribute should not be advertised to BGP peers outside a confederation boundary.

none Removes all communities from BGP routes serviced by this route map.

> *See Table 1 on page 30 for a summary of the well-known communities (for example, INTERNET, NO_EXPORT_SUBCONFED, etc.) and the policy expected for these communities.*

Attributes, such as local preference and metric, can also be defined for the community string. A discussion of the **set** commands used to define these community attributes begin with *A. Prepending Private AS Numbers for Load Balancing on page 43*.

Step 3: Enabling an AOS Device to Send or Receive BGP Communities

A BGP AF neighbor must be configured for BGP communities before it is able to send or receive a community attribute. Use the **send-community standard** command to enable this peer to accept a community attribute and add the community attribute to any advertisement sent by this peer. This command is issued from the BGP AF Neighbor Configuration mode. To enable the AF neighbor on the default VRF to accept community attributes, enter the command as follows:

```
(config-bgp-ipv4-neighbor)#send-community standard
```

To enable the AF neighbor on a nondefault (named) VRF to accept community attributes, enter the command as follows:

```
(config-bgp-vrf-ipv4-neighbor)#send-community standard
```

Step 4: Deleting Communities from a Route

BGP communities are not completely standardized. External neighbors could place routes in a community for which the local network does not define a policy or for which it defines a substantially different policy. A network administrator may not want to apply the policies that the external neighbor is requesting with the community attribute. As a result, the administrator may need to remove certain communities from inbound routes in order to enforce the organization’s policies.

Deleting communities from a route can be accomplished by first creating a community list that *permits* the communities that are to be deleted (refer to *Step 1: Configuring a Community List on page 30*). Next, create a route map (refer to *Route Maps on page 37*) and use the **set comm-list delete** command to specify a list of communities to delete:

```
(config-route-map)#set comm-list <name> delete
```

<name> Specifies the name of the community list that contains the communities to delete.

Apply the route map to the AF neighbor as an inbound policy (refer to *Applying a Route Map Entry to a BGP Neighbor on page 46*).

If a network defines local communities, the administrator may need to remove these from the routes before the local router advertises the routes to an external neighbor. This scenario requires the administrator to configure a community list that permits the local communities. The community list is then matched to a route map entry and the route map is applied to the BGP AF neighbor as an outbound policy.

> NOTE *If a route map is already in place to set policies or filter routes, this route map should also be used to delete the specified communities. Each route map sequence number entry that could potentially let in a route should be evaluated to determine if deleting communities also applies. If so, the **set comm-list** <name> **delete** command should be added for each applicable route map sequence number. This is because the router stops processing a route map as soon as it finds a match. If a separate, earlier entry permits a route (as described in the preceding paragraphs), the router will immediately add the route to the BGP database without applying any policies that are set in later entries.*

Distribute List

Use the **distribute-list** command to add route filtering functionality by assigning inbound and outbound access control lists (ACLs) to an AF neighbor. Only one inbound/outbound pair of ACLs can be configured for a particular neighbor. To apply an IPv4 ACL filter to the IPv4 AF neighbor on the default VRF, enter the command as follows:

(config-bgp-ipv4-neighbor)#**distribute-list** <*ipv4 acl name*> **[in | out]**

<*ipv4 acl name*> Specifies an IPv4 ACL name. This is a standard or extended IPv4 ACL against which the contents of the incoming/outgoing routing updates are matched.

in Applies route filtering to inbound data.

out Applies route filtering to outbound data.

To apply an IPv4 ACL filter to the AF neighbor on a nondefault (named) VRF, enter the command as follows:

(config-bgp-vrf-ipv4-neighbor)#**distribute-list** <*ipv4 acl name*> **[in | out]**

> NOTE *Refer to IP ACLs in AOS (ADTRAN's Knowledge Base article 3087) for information on how to create standard or extended IPv4 ACLs.*

To apply an IPv6 ACL filter to the IPv6 AF neighbor on the default VRF, enter the command as follows:

(config-bgp-ipv6-neighbor)#**distribute-list** <*ipv6 acl name*> **[in | out]**

<*ipv6 acl name*> Specifies an IPv6 ACL name. This is a standard or extended IPv6 ACL against which the contents of the incoming/outgoing routing updates are matched.

in Applies route filtering to inbound data.

out Applies route filtering to outbound data.

To apply an IPv6 ACL filter to the AF neighbor on a nondefault (named) VRF, enter the command as follows:

(config-bgp-vrf-ipv6-neighbor)#**distribute-list** <*ipv6 acl name*> **[in | out]**

NOTE *Refer to Using IPv6 in AOS (ADTRAN's Knowledge Base article 3505) for information on how to create standard or extended IPv4 ACLs.*

Next-Hop Self

Use the **next-hop-self** command to force the NEXT_HOP attribute to be changed to this unit's address for each network it advertises to the neighbor address. To set the NEXT_HOP attribute for the IPv4 AF neighbor on the default VRF, enter the command as follows:

(config-bgp-ipv4-neighbor)#**next-hop-self**

To set the NEXT_HOP attribute for the IPv4 AF neighbor on the nondefault (named) VRF, enter the command as follows:

(config-bgp-vrf-ipv4-neighbor)#**next-hop-self**

IGPs, such as RIP and OSPF, always use the source IP address of a routing update as the next-hop address for each network that is placed in the routing table. Conversely, since BGP routes AS-to-AS, the default next hop that is advertised is the next AS. This behavior can present a problem in situations where an iBGP router learns about networks outside of its AS through one of its iBGP peers. By default, the next-hop address for the external networks advertised to the iBGP router is the entry point for the next AS. When the iBGP router receives packets destined for one of the external networks, it performs a recursive lookup of the entries in its own IGP routing table to determine how to reach the BGP next-hop address. Unless the iBGP router has a static route or an entry in its IGP routing table indicating how to reach the edge router in the external AS, packets destined for those networks will be dropped. A remedy for this scenario is for the iBGP peer to advertise its own IP address as the next-hop address to the external networks. Consider the following example:

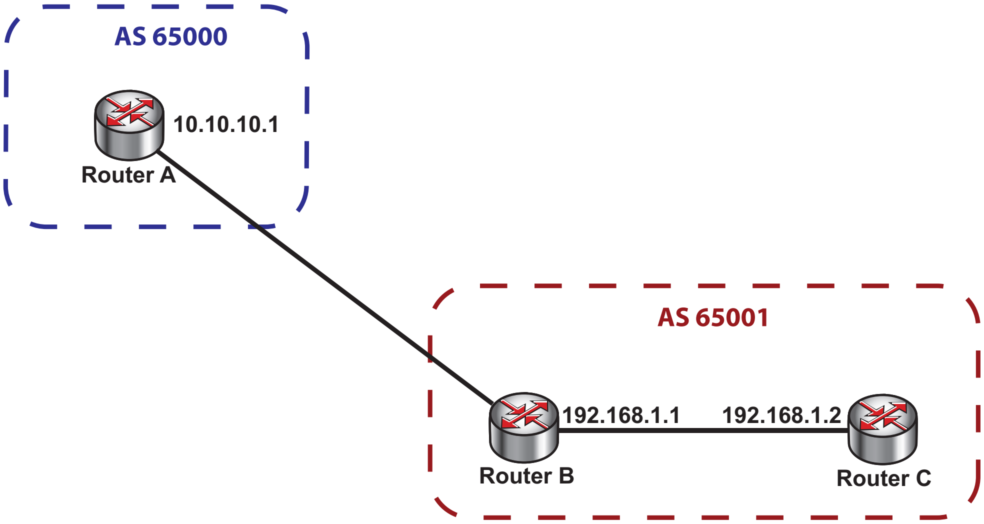


Figure 3. Using the Next-Hop-Self Command between iBGP Peers

Router B in AS 65001 has an eBGP neighbor relationship with Router A in AS 65000. All networks in AS 65000 are advertised from Router A to Router B with a next-hop IPv4 address of 10.10.10.1. Subsequently, when Router B announces these networks to its iBGP neighbor (Router C), the BGP default setting is to announce that the next hop to reach these networks is the entrance to AS 65000 (10.10.10.1). Router C MUST have either a static route or an entry in its IGP routing table, indicating a route to reach the edge router at 10.10.10.1. Otherwise, any information destined for networks in AS 65000 will be dropped by Router C. The **next-hop-self** command can be issued on Router B so that Router B's IPv4 address (192.168.1.1) is advertised to Router C as the next-hop address for networks in AS 65000. Since Router B and C are directly connected, Router C's routing table contains a route to Router B.

> NOTE *Example 3 on page 50 further demonstrates the use of the* ***next-hop-self*** *command.*

No Default Originate

The **no default-originate command** prevents the unit from sending the default route to an AF neighbor. This is the default setting for AOS devices. To prevent the IPv4 AF neighbor on the default VRF from sending the default route, enter the command as follows:

```
(config-bgp-ipv4-neighbor)#no default-originate
```

To prevent the IPv4 AF neighbor on a nondefault (named) VRF from sending the default route, enter the command as follows:

```
(config-bgp-vrf-ipv4-neighbor)#no default-originate
```

> NOTE *This command cannot be enabled. In other words, the command* ***default-originate*** *cannot be entered to enable the default route to be sent to a BGP neighbor.*

The transmission of default routes to AF neighbors is accomplished by manually entering a default route in the AF configuration. This is done by entering the following command for an IPv4 or IPv6 AF (from either the default VRF BGP AF Configuration mode or the nondefault (named) VRF BGP AF Configuration mode):

```
(config-bgp-ipv4)#network 0.0.0.0 mask 0.0.0.0
OR
(config-bgp-ipv6)#network 2001:DB8:3F::/64
```

> NOTE *An entry for the default route must appear in the IGP routing table in order for the previous command to work in BGP. Refer to Example 6 on page 59 for an illustration using the* ***network*** *command to enable transmission of the default route to the eBGP neighbor.*

Prefix List

Prefix lists are used in BGP configurations to define the routes that a router can advertise to or receive from an AF neighbor. Common uses for prefix lists include:

- Preventing a network from becoming a transit for external traffic when multihoming
- Receiving only routes from remote VPN sites

- Prohibiting the advertisement of a network
- Load balancing outbound traffic

IP address, prefix length, or other attributes are defined in a prefix list before it can be assigned to an AF neighbor. To configure a prefix list, and apply it an AF neighbor, follow these steps:

> *Refer to Example 2 on page 49 for an example using prefix lists to filter routes.*

Step 1: Creating the Prefix List (Specifying the Prefix to be Matched)

First, use either the **ip prefix-list** (IPv4) or **ipv6 prefix-list** (IPv6) command from the Global Configuration mode to specify a prefix to be matched or a range of mask lengths:

```
(config)#ip prefix-list <name> seq <number> [deny | permit] <network ip/length>
(config)#ip prefix-list <name> seq <number> [deny | permit] <network ip/length> [ge | le] <value>
```

OR

```
(config)#ipv6 prefix-list <name> seq <number> [deny | permit] <ipv6 address/prefix-length>
(config)#ipv6 prefix-list <name> seq <number> [deny | permit] <ipv6 address/prefix-length> [ge | le]
    <value>
```

<name> Specifies the name of the list. Up to **80** characters are allowed in a name.

<number> Specifies the entry's unique sequence number that determines the processing order. Lower numbered entries are processed first. Range is **1** to **4294967294**.

permit *<network ip/length>* Permits access to entries matching the specified network IPv4 address and the corresponding network prefix length (for example, **10.10.10.0/24**).

deny *<network ip/length>* Denies access to entries matching the specified network IPv4 address and the corresponding network prefix length (for example, **10.10.10.0/24**).

<ipv6 address/prefix-length> Specifies a network IPv6 address and the corresponding network prefix length. IPv6 address and prefixes are expressed in colon hexadecimal format (for example, **2001:DB8:0:3F3B::/64**).

le *<value>* Specifies the upper end of the range. Range is **0** to **32**.
ge *<value>* Specifies the lower end of the range. Range is **0** to **32**.

> *If the network address is entered without specifying a range for prefix lengths, the router assumes that the route must be an exact match. For example, if the command* ***ip prefix-list TEST seq 5 permit 10.1.0.0/16*** *is entered, the BGP interface will only accept routes to the entire 10.1.0.0 /16 subnet. It will not accept routes to a network, such as 10.1.1.0 /24, which was subdivided from the /16 network.*

Routes to subnets within the larger network can be permitted or denied by specifying the permitted range of prefix lengths. For example, the filter could allow all routes to subnets in the 10.1.0.0 /16 network with a prefix length up to and including 24:

(config)#**ip prefix-list TEST seq 5 permit 10.1.0.0/16 ge 16 le 24**

NOTE *The **ge** keyword indicates that the length must be **greater than or equal to** that specified in order to match. The **le** keyword indicates that the length must be **less than or equal to** that specified in order to match. If **ge** is only specified, the router assumes 32 as the upper limit. If **le** is only specified, the router assumes the network address's length as the lower limit.*

A filter that exactly matches a prefix length can be created by entering the length for both the **ge** and **le** values. For example, the filter could allow any routes to a /24 subnet in the 10.1.0.0 /16 range, but not accept a route to the entire 10.1.0.0 /16 network:

(config)#**ip prefix-list TEST seq 5 permit 10.1.0.0/16 ge 24 le 24**

Step 2: Assigning the Prefix List to an AF Neighbor

After a prefix list has been defined, use the **prefix-list** <*name*> command to assign the prefix list to an AF neighbor, on either the default or nondefault (named) VRF, and specify whether the list will be used to filter inbound or outbound routes. To assign the prefix list to an IPv4 AF neighbor on the default VRF, enter the command as follows:

(config-bgp-ipv4-neighbor)#**prefix-list** <*name*> **[in | out]**

<*name*> Assigns the specified prefix list to this BGP neighbor.

in Specifies that all inbound BGP route updates received from the neighbor are filtered.

out Specifies that all outbound BGP route updates being sent to the neighbor are filtered.

To assign the prefix list to an IPv4 AF neighbor on a nondefault (named) VRF, enter the command as follows:

(config-bgp-vrf-ipv4-neighbor)#**prefix-list** <*name*> **[in | out]**

To assign the prefix list to an IPv6 AF neighbor on the default or nondefault VRF instance, enter the **prefix-list** <*name*> **[in | out]** command from the IPv6 AF neighbor's configuration mode. The prefix list specified by the <*name*> parameter should be an IPv6 prefix list created with the **ipv6 prefix-list** command.

NOTE *A prefix list can be used to create even more complicated policies when it is applied to a route map entry rather than a BGP neighbor (as shown above). Refer to Route Maps on page 37 for more information on this option.*

NOTE *You cannot assign a prefix list of one type to the AF of another type. For example, you cannot assign an IPv4 prefix list to an IPv6 AF neighbor.*

Route Maps

Route maps allow configuration of more complex policies than prefix lists. In addition to filtering routes according to network address and prefix length, routes can be filtered according to their AS path, metric value, or BGP community. A BGP community is a group of routes to which a BGP router applies the same policies. Refer to *BGP Communities on page 29* for more information on BGP communities.

Route maps can also be configured to apply various attributes to the routes it filters. A route map applied to outbound data determines how the router advertises routes to a neighbor. The **outbound** route map can be configured to perform such tasks as:

- Define the routes that the router can advertise according to specified attributes or prefixes
- Prepend private AS numbers to specific routes to help balance inbound traffic
- Set an MED on specific routes to help balance inbound traffic
- Request that the neighbor advertise the route to certain communities only

When a route map is applied to inbound data, it determines which of the service provider's advertised routes the local router accepts. The **inbound** route map can be configured to perform such tasks as:

- Filter external routes according to specified attributes or prefixes
- Apply attributes to filtered routes, including: local preference, community, MED value, and prepended AS path
- Delete communities defined for the routes

The route map itself is created first. Matching criteria and attributes are defined within the route map configuration menu. Once a route map has been established, it can be assigned to a BGP neighbor. To create a route map, follow these steps:

Step 1: Creating a Route Map

Use the **route-map** command to create a route map and enter the Route Map Configuration mode.

```
(config)#route-map <name> <number>
(config)#route-map <name> [deny | permit] <number>
```

<name> Specifies a name for the route map.

deny Specifies not to process routes matching the specified route map attributes.

permit Processes routes matching the specified route map attributes.

<number> Specifies a sequence number for this route entry. Range is **1** to **4294967295**.

> NOTE *After creating a route map, route map attributes can be defined from the Route Map Configuration mode. Enter* ***?*** *at the* ***(config-route-map)#*** *prompt to explore the available options.*

Step 2: Defining Routes and Attributes to be Advertised and Filtered

Define the routes and attributes to be advertised outbound or filtered inbound.The advertisements sent from a BGP interface to a neighbor or received by a BGP interface from a neighbor can be controlled according to the route's:

- Network address and prefix length
- AS path
- Community
- Metric

Routes that the BGP interface will advertise outbound or filter inbound are selected by entering a **match** command in a route map entry. The difference between an inbound filter and an outbound filter is seen when the route map is applied to a BGP neighbor as an **inbound** policy rather than an **outbound** policy. Refer to *Applying a Route Map Entry to a BGP Neighbor on page 46*. For a list of available filtering options for BGP, see *Table 2 on page 39*.

Table 2. Defining Routes to Be Filtered

Filtering According To	Command Syntax
Network address and/or prefix length	**match ip address prefix-list** <*name*> **match ipv6 address prefix-list** <*name*>
ACL	**match ip address** <*ipv4 acl name*> **match ipv6 address** <*ipv6 acl name*>
AS_PATH	**match as-path** <*name*>
Community	**match community** <*name*> **exact-match**
Metric	**match metric** <*value*>

Detailed explanations of the above **match** commands begin with *A. Filtering Routes According to Network Address on page 40*.

If a BGP route does not contain a qualifying prefix or attribute that matches any of the filters specified in the route map or if a BGP route matches a deny route map entry, then the route will not be allowed in or out.

If the only action required is filtering of routes, then the ***match*** *command is entered and the route map is applied to the BGP neighbor as either an outbound or inbound policy (refer to Next-Hop Self on page 34).*

If an attribute is to be applied to the route, then a **set** command must be entered in addition to the **match** command. Attributes are applied to the routes selected by the **match** command. The following attributes can be applied to inbound filtered or outbound advertised routes:

- Community
- Prepend AS path
- MED metric
- Local preference
- Delete a community

Detailed explanations of the **set** command attributes begin with *Step 3: Applying Filters to the Route Map on page 43*.

A. Filtering Routes According to Network Address

One way to use route maps to filter routes is according to the network address and/or prefix length. A prefix list is first created to define the routes that are to be filtered by the BGP interface (refer to *Prefix List on page 35*). The prefix list delineates either routes that the BGP interface will advertise outbound or inbound routes that should be filtered. An exact route can be specified or a range of prefix lengths for routes to variable length subnets. After the prefix list has been configured, it is referenced in a route map entry. The route map entry is then applied to a BGP neighbor (refer to *Applying a Route Map Entry to a BGP Neighbor on page 46*).

Use the **match ip address prefix-list** command to configure the route map to route traffic based on an IPv4 prefix list route filter:

```
(config-route-map)#match ip address prefix-list <name>
```

<name> Specifies the name of the prefix list.

Use the **match ipv6 address prefix-list** command to configure the route map to route traffic based on an IPv6 prefix list route filter:

```
(config-route-map)#match ipv6 address prefix-list <name>
```

Another way to use route maps to filter routes according to network address is by using standard IP ACLs. As with prefix lists, an ACL is first created to define the routes that are to be filtered by the BGP interface. Refer to *IP ACLs in AOS* (ADTRAN's Knowledge Base article 3087) for information on how to create a standard IPv4 ACL, or to *Using IPv6 in AOS* (ADTRAN's Knowledge Base article 3505) for information on how to create a standard IPv6 ACL. After the ACL has been configured, it is referenced in a route map entry (see below) or applied to an AF neighbor using the **distribute-list** command (refer to *Distribute List on page 33*). The route map entry is then applied to an AF neighbor (refer to *Applying a Route Map Entry to a BGP Neighbor on page 46*).

Use the **match ip address** command to configure the route map to process traffic based on an IPv4 ACL name defined with the **ip access-list** command:

```
(config-route-map)#match ip address <ipv4 acl name>
```

<ipv4 acl name> Specifies the name of the IPv4 ACL to match.

Use the **match ipv6 address** command to configure the route map to process traffic based on an IPv6 ACL name defined with the **ipv6 access-list** command:

```
(config-route-map)#match ipv6 address <ipv6 acl name>
```

<ipv6 acl name> Specifies the name of the IPv6 ACL to match.

*Use **set** commands to configure any attributes (prepend AS_PATH,* MULTI_EXIT_DISC, *LOCAL_PREF, etc.) to be applied to the routes prior to associating the route map with the BGP neighbor. Refer to Step 3: Applying Filters to the Route Map on page 43 for detailed explanations of the **set** command attributes.*

Refer to Example 5 on page 55 for a detailed BGP configuration example featuring the use of route maps to filter routes according to network address.

B. Filtering Routes According to AS Path

Routes can also be filtered according to the traversed autonomous systems listed in the AS_PATH field. For advertised routes, this type of filtering allows a degree of influence over which autonomous systems external neighbors can access. For example, service provider routers can filter routes with paths that include customer AS numbers to prevent themselves from advertising private customer routes to unauthorized peers. Private networks do not typically transmit traffic from AS to AS. Therefore, filtering advertised routes according to AS path is not usually necessary when configuring eBGP in a private network.

A list of the AS paths to be filtered is created first. The AS path list is then referenced in a route map entry to define the paths to be filtered. Lastly, the route map is applied to an AF neighbor (refer to *Applying a Route Map Entry to a BGP Neighbor on page 46*).

Use the **as-path-list** command to create AS path lists for route map use:

(config)#**as-path-list** <*name*>

<*name*> Specifies the name of the AS path list.

Next, specify the AS numbers to be filtered and specify whether the routes containing these AS numbers should be permitted or denied:

(config-as-path-list)#**[deny | permit]** <*value*>

<*value*> Specifies permitting or denying routes that contain this value in their AS_PATH attribute. This is a string that follows the format of AS regular expressions to filter an AS path. Refer to *AS Regular Expressions on page 110* for a detailed list of valid AS regular expressions.

The AS path list is only compared against the AS_PATH attribute on a BGP prefix, which is also displayed in the output of the **show bgp** command. For example, a router can be permitted to advertise only routes that use both AS 200 and AS 400:

```
(config-as-path-list)#permit (\b200\b.*400\b)|(\b400\b.*200\b)
```

However, the statement above only permits routes that use both AS 200 and AS 400. Permit any routes that use either AS by entering separate statements:

```
(config-as-path-list)#permit \b200\b
(config-as-path-list)#permit \b400\b
```

Permitting AS number 200 selects any routes that include that value, even if the AS field also includes other values. In other words, entering **permit \b200\b** permits routes containing AS 200, as well as AS 200 and AS 400, while entering **permit (\b200\b.*400\b)|(\b400\b.*200\b)** only permits routes containing both AS 200 and AS 400. Therefore, it may be necessary to explicitly deny any values that should not be included in the field.

Another example might be where the router is allowed to advertise routes that use AS 200 or AS 400, but not routes that force traffic to travel through both AS 200 and AS 400:

```
(config-as-path-list)#deny (\b200\b.*400\b)|(\b400\b.*200\b)
(config-as-path-list)#permit \b200\b
(config-as-path-list)#permit \b400\b
```

NOTE *It is important to enter any **deny** commands before the **permit** commands since the router processes statements in the AS path list in the order that they are entered.*

After configuring the AS path list, use the **match** command to reference the list in a route map entry.

```
(config-route-map)#match as-path <name>
```

<name> Specifies the name of the AS path list.

NOTE *Use **set** commands to configure any attributes (prepend AS_PATH,* MULTI_EXIT_DISC, *LOCAL_PREF, etc.) to be applied to the routes prior to applying the route map to the BGP neighbor. Refer to Step 3: Applying Filters to the Route Map on page 43 for detailed explanations of the **set** command attributes.*

The route map is then applied to an AF neighbor (refer to *Applying a Route Map Entry to a BGP Neighbor on page 46* for more information).

C. Filtering Routes According to Community

If a network places routes in communities, the routes that the local router advertises can be filtered according to these communities. The first step to filtering routes according to community is to create a community list (refer to *Step 1: Configuring a Community List on page 30*) that either permits or denies BGP routes based on well-known or privately defined BGP communities. Next, unless previously configured, a route map must be created (refer to *Route Maps on page 37*). Use the **match** command to reference the community list in the route map entry:

```
(config-route-map)#match community <name> exact-match
```

<name> Specifies the name of the community list.

exact-match Optional. Specifies that the route map must match the community name exactly. When the **exact-match** keyword is used, the entire community string must be defined as advertised for it to qualify as an exact match. Without this keyword, matches will result if the defined string appears *anywhere* in the community field.

This command does not define a community for routes. It selects routes according to their predefined community or communities. Other BGP neighbors, either internal or external, should have placed the route in a community.

*Use **set** commands to configure any policies (prepend AS_PATH,* MULTI_EXIT_DISC, *LOCAL_PREF, etc.) to be applied to the community. Refer to Step 3: Applying Filters to the Route Map on page 43 for detailed explanations of the **set** command attributes.*

Lastly, the route map should be applied to an AF neighbor as an outbound or inbound policy (refer to *Applying a Route Map Entry to a BGP Neighbor on page 46*).

Step 3: Applying Filters to the Route Map

After deciding how to filter routes, apply these filters to the route map using the **set** commands. These commands and their use are described in the following sections.

A. Prepending Private AS Numbers for Load Balancing

A router sends identical routes to all neighbors unless policies are configured to filter and add attributes to the routes. When service provider routers receive multiple identical routes from an organization, it is up to the service provider to select the connection over which inbound traffic is sent to the organization. The customer can attempt to load balance inbound traffic over multiple Internet connections by influencing the service provider routers' selection process. One way to accomplish this is to prepend extra hops in the AS path of certain routes. For example, a router has two connections to the Internet: one to Service Provider A and one to Service Provider B. Inbound traffic always arrives over the connection from Service Provider A. Several fabricated AS hops can be prepended to the routes for half of the private networks sent from the router to Service Provider A. The advertisements containing extra AS hops would make the service provider routers more likely to route traffic destined to these networks through Service Provider B.

To prepend AS hops to a route, a route map is created and the **match** command is used to select the routes to which the router should prepend the AS hops. Generally, routes are selected according to their network address and prefix length. However, routes can be selected according to other attributes as well.

*Refer to Route Maps on page 37 for information on how to create a route map. Refer to A. Filtering Routes According to Network Address on page 40 for information on the **match** command used to select routes according to network address and prefix length.*

Use the **set as-path prepend** command to prepend the hops to the selected routes. The router can be configured to prepend one or more fabricated AS hops to the selected routes.

```
(config-route-map)#set as-path prepend <number>
```

<number> Specifies a number to be prepended to the AS_PATH value as an AS number. Valid range is **1** to **4294967295**.

Alternatively, the router can simply repeat the last AS in a route up to ten times.

(config-route-map)#**set as-path prepend last-as** <*number*>

<*number*> Specifies the number of times to repeat the last AS. Valid range is **1** to **10**.

> NOTE
>
> *Be sure to consult with the service provider before prepending any fabricated AS numbers to a path. It is important to ensure that the fabricated AS path does not conflict with route policies that the service provider router implements. It is also important to discuss with the service provider what BGP attributes they consider when making routing decisions. This information will verify whether AS_PATH is a valid way to influence traffic to your network.*

B. Setting a MED Metric

Another way to influence neighbors to select a certain connection for inbound traffic is to set different metrics on the routes that are sent to separate neighbors. Since BGP prefers routes with a lower metric, the connection to the neighbor that receives the route with the lowest metric is more likely to be selected.

> NOTE
>
> *The algorithm BGP uses to select routes relies on many factors, some of which are dependent upon configurations on the remote router. It is impossible to ensure that the route with the lower metric will actually be selected.*

This metric is sometimes called the multi-exit discriminator or MED because it is used to differentiate routes sent over various external connections to the same neighboring AS.

When MEDs are used, routes to a specific part of the network are typically classified according to their destination address. This classification is accomplished using one prefix list or several; depending on the network setup and the goal. Refer to *Prefix List on page 35* for information on how to create prefix lists. Separate route maps are then configured for each neighbor to which the router connects. Refer to *Route Maps on page 37* for information on how to create a route map. A prefix list is associated with each route map entry. Again, depending on the network setup and the goal, the same prefix list can be associated with each route map entry or there can be a different prefix list associated with every route map entry.

Within each route map entry, use the **set metric** command to specify a metric value for the routes that have been selected:

(config-route-map)#**set metric** <*value*>

<*value*> Sets the metric value. Valid range is **0** to **4294967295**.

> NOTE
>
> *The route with the lowest MULTI_EXIT_DISC value is preferred in BGP. Refer to Multi-Exit Discriminators (MEDs) on page 19 for information on options available in AOS for handling MEDs received.*

> NOTE
>
> *Refer to Example 5 on page 55 for a detailed BGP configuration example featuring the use of the MED metric to influence which path is selected for inbound traffic to a local network.*

NOTE *When specifying a MULTI_EXIT_DISC value, the attribute should be applied **outbound** to a BGP neighbor. Refer to Applying a Route Map Entry to a BGP Neighbor on page 46 for additional information.*

C. Setting Local Preference for Inbound Routes

The local preference attribute can be used to influence the best path used to transmit information from a local network to a private remote network. Adjusting the local preference value on inbound routes from a remote network can affect the local router's decision when transmitting traffic outbound to the remote network.

NOTE *The local preference attribute can be set for outbound routes, but it is only relevant in iBGP scenarios because the local preference attribute is not retained across different autonomous systems.*

Use the **set local-preference** command to change the LOCAL_PREF value for selected inbound routes:

```
(config-route-map)#set local-preference <value>
```

<*value*> Sets the local preference value. Valid range is **0** to **4294967295**.

NOTE *The default local preference for all BGP routes in AOS is **100**. The default value can be changed using the **bgp default local-preference** <value> command. The valid range for the default value is **0** to **4294967295**.*

NOTE *The route with the largest local preference value is preferred in BGP.*

NOTE *Refer to Example 5 on page 55 for a detailed BGP configuration example featuring the use of local preference on inbound routes to influence which path is selected for outbound traffic to a private remote network.*

D. Deleting a Community List

Use the **set comm-list delete** command to specify a list of communities to delete from the route:

```
(config-route-map)#set comm-list <name> delete
```

<*name*> Specifies the name of the community list that contains the list of community strings to delete.

NOTE *A community list must be defined using the **community-list** command before the **set comm-list delete** command can be used. Refer to Step 1: Configuring a Community List on page 30 for detailed information on configuring community lists.*

Applying a Route Map Entry to a BGP Neighbor

After a route map entry has been configured, it must be applied to a BGP neighbor. Use the **route-map** command in BGP AF Neighbor Configuration mode to assign the route map to a specific neighbor. To apply the route map to an IPv4 AF neighbor in the default VRF, enter the command as follows:

(config-bgp-ipv4-neighbor)#**route-map** <*name*> **[in | out]**

<*name*> Assigns the specified route map to this BGP neighbor.

in Specifies the filtering/modification of all inbound BGP route updates (for filtering external routes and setting inbound policies).

out Specifies the filtering/modification of all outbound BGP route updates (for advertising routes to the external neighbor and setting outbound policies).

To apply the route map to an IPv4 AF neighbor in a nondefault (named) VRF, enter the command as follows:

(config-bgp-vrf-ipv4-neighbor)#**route-map** <*name*> **[in | out]**

NOTE *Before a route map can be assigned to a BGP neighbor, it must first be defined using the* ***route-map*** *command in the Global Configuration mode (refer to Route Maps on page 37).*

NOTE *Refer to Example 5 on page 55 for a detailed BGP configuration example where route maps are applied inbound and outbound to different BGP neighbors.*

Soft Reconfiguration Inbound

Soft reconfiguration enables the AOS unit to store all updates from a neighbor in case the inbound policy is changed. The command is issued in BGP AF Neighbor Configuration mode and allows a network administrator to reconfigure BGP policies without clearing active BGP sessions. Administrators can then institute new policies at any time without forcing the neighbors to reestablish their connection and possibly disrupt traffic.

BGP updates are stored prior to filtering; thus, allowing the **clear bgp soft** command to be used in the absence of route refresh (RFC 2918) capability. The unfiltered table is used when an inbound policy is changed; allowing the AOS device to implement policy changes immediately based on the stored table instead of having to wait on a new table to be built after a hard reset. A soft reset is beneficial over a hard reset because it allows policy updates without disrupting network traffic flow. A hard reset terminates the existing BGP session, effectively removing all routes learned from a neighbor. A new session is then created and all of the routes must be relearned. Because this process takes place with a hard reset, a network outage can potentiality occur until the BGP database and route table have been rebuilt.

NOTE *Refer to Clear BGP on page 100 for more information on this command.*

Use the **soft-reconfiguration inbound** command to enable the AOS device to store BGP updates for the specified AF neighbor. By default, this command is disabled. To enable soft reconfiguration on an IPv4 BGP AF neighbor on the default VRF , enter the command as follows:

```
(config-bgp-ipv4-neighbor)#soft-reconfiguration inbound
```

BGP Example Configurations

The example scenarios contained within this section are designed to enhance understanding of BGP configurations on AOS products. The examples describe some of the common real-world applications of BGP. All configurations provided in this section use the command line interface (CLI).

> *The configuration parameters entered in these examples are sample configurations only. These applications should be configured in a manner consistent with the needs of your particular network. CLI prompts have been removed from the configuration examples to provide a method of copying and pasting configurations directly from this configuration guide into the CLI. These configurations should not be copied without first making the necessary adjustments to ensure they will function properly in your network.*

Some commands shown in the example configurations in this guide are already enabled as the default setting in the unit. These commands will not appear in the output when the **show running-config** command is issued. Issue the **show run verbose** command to see all commands (including those that do not appear when the **show running-config** command is issued).

Example 1: MPLS Basic Setup for Private Internet Protocol (PIP)

This example illustrates a typical PIP setup where several remote sites are connected by an MPLS provider. The local AOS router is acting as the customer edge (CE) router and will form a neighbor relationship with the provider edge (PE) router to exchange IPv4 BGP routes over a Point-to-Point Protocol (PPP) connection. The PE router will learn all of the other remote customer subnets (192.168.2.0 /24 and 192.168.3.0 /24) using BGP and advertise them to the local AOS router. The local AOS router will have an IPv4 static default route to a firewall on the local area network (LAN) for Internet access.

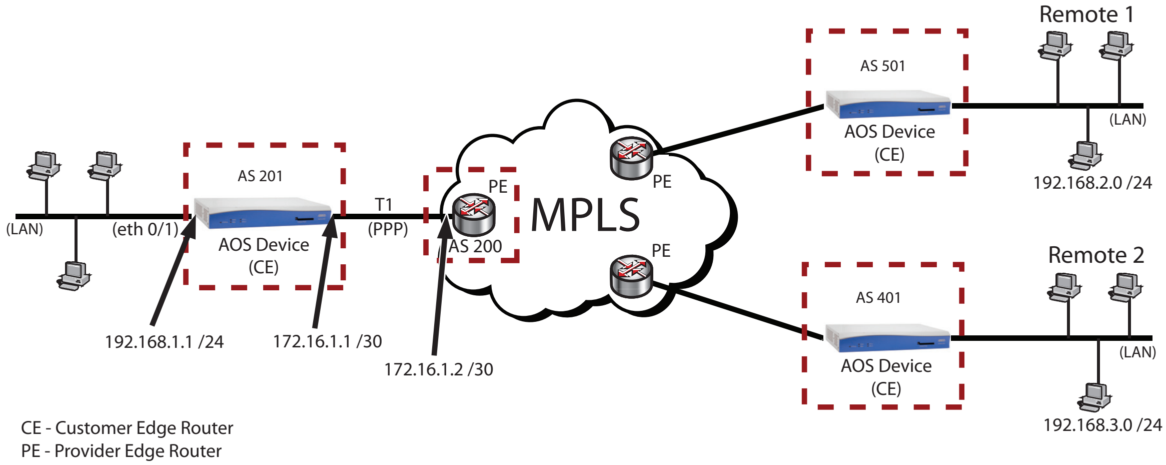


Figure 4. Typical IPv4 PIP Application

The following configuration applies to Example 1:

```
!
interface eth 0/1
    ip address 192.168.1.1 255.255.255.0
    no shutdown
!
interface t1 1/1
    clock source line
    tdm-group 1 timeslots 1-24 speed 64
    no shutdown
!
interface ppp 1
    ip address 172.16.1.1 255.255.255.252
    no shutdown
    cross-connect 1 t1 1/1 1 ppp 1
!
router bgp 201
    neighbor 172.16.1.2
        remote-as 200
        no shutdown
        exit
    address-family ipv4
        network 192.168.1.0 mask 255.255.255.0
        neighbor 172.16.1.2
            soft-reconfiguration inbound
            no shutdown
            exit
        exit
    !
ip route 0.0.0.0 0.0.0.0 192.168.1.254
!
```

NOTE *Depending on the provisioning from the MPLS provider, a default route for Internet access may be advertised through BGP.*

NOTE *When the **eth 0/1** interface is up, there will be a directly connected route for the 192.168.1.0 /24 subnet in the routing table. The route will allow this network to be advertised in BGP using the network command. When the interface is not up, the route will not be advertised in BGP because it is not in the routing table of the router.*

NOTE *The command **clock source line** is enabled by default. Therefore, this command will not appear in the output when the **show running-config** command is issued.*

Example 2: Filtering Routes with Prefix Lists

BGP routes advertised and received on an interface can be filtered using prefix lists. The following example illustrates the use of prefix lists to discard incoming routing information and to limit the routes advertised to certain peers. The AOS device in *Figure 5* is expecting a specific route from its IPv4 eBGP neighbor (208.61.209.253). All other advertised routes from this neighbor are to be discarded. A prefix list (EXPECTED-ROUTE) is used to define the specific IPv4 subnet (208.61.209.0 /29) the AOS device is expecting from the eBGP neighbor. The implicit deny at the end of a prefix list denies all other routes. This prefix list is applied inbound from the eBGP neighbor.

All BGP routes learned from one neighbor are advertised to all other BGP neighbors by default. The customer wants to advertise a specific route from the AOS device to the IPv4 eBGP neighbor (208.61.209.253) and at the same time prevent the eBGP neighbor from learning about other BGP routes advertised to the AOS device from its iBGP neighbor (65.162.109.202). A prefix list (ADVERTISE) is used to define the specific IPv4 route (65.162.109.201 /29) that is to be advertised from the AOS device to the eBGP neighbor. The implicit deny at the end of the prefix list will prevent any other BGP routes from being advertised on the link. The prefix list is applied outbound toward the eBGP neighbor.

Lastly, it is desired that the AOS device in *Figure 5* learn routes from its iBGP neighbor (65.162.109.202), but not advertise any routes to this neighbor. A prefix list (FILTER) is used to create a *deny all* statement. The prefix list is applied outbound toward the iBGP neighbor.

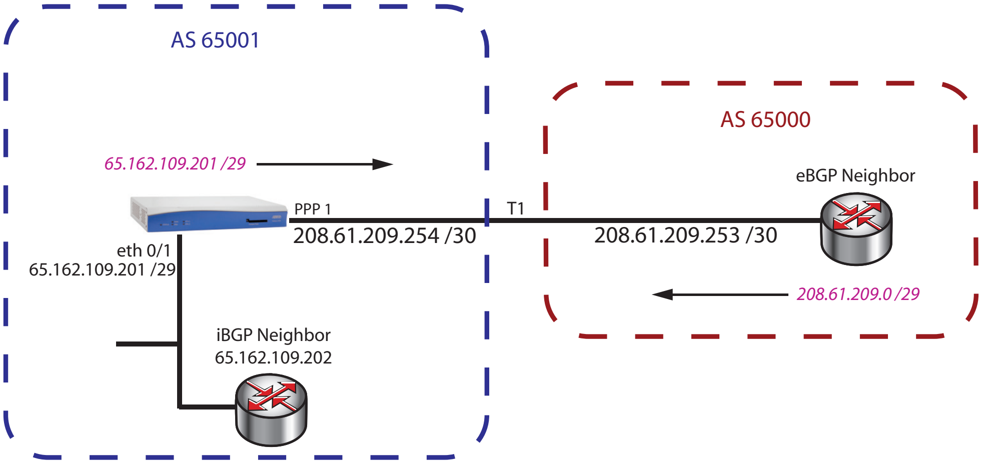


Figure 5. Using Prefix Lists to Filter IPv4 Routes Sent to and Received from BGP Neighbors

The following configuration applies to Example 3:

```
!
interface eth 0/1
    ip address  65.162.109.201 255.255.255.248
    no shutdown
!
interface t1 1/1
     clock source line
     tdm-group 1 timeslots 1-24 speed 64
     no shutdown
!
```

```
interface ppp 1
     ip address  208.61.209.254 255.255.255.252
     no shutdown
     cross-connect 1 t1 1/1 1 ppp 1
!
ip prefix-list ADVERTISE seq 10 permit 65.162.109.201/29
ip prefix-list EXPECTED-ROUTE seq 10 permit 208.61.209.0/29
ip prefix-list FILTER seq 10 deny 0.0.0.0/0 le 32
!
router bgp 65001
     neighbor 658.162.109.202
        remote-as 65001
        no shutdown
        exit
    neighbor 208.61.209.253
        remote-as 65000
        no shutdown
        exit
    address-family ipv4
        neighbor 65.162.109.202
            prefix-list FILTER out
            soft-reconfiguration inbound
            exit
        neighbor 208.61.209.253
            prefix-list EXPECTED-ROUTE in
            prefix-list ADVERTISE out
            soft-reconfiguration inbound
            exit
         exit
    !
!
```

Example 3: Multihoming and Influencing Traffic over a Preferred Path

Multihoming is when a router has more than one connection to the Internet. The following example illustrates a customer multihoming to two different service providers. The customer owns an IPv4 public block 208.61.209.0 /29 that will be advertised to both service providers. The preferred path for incoming traffic is the high speed Metro-Ethernet connection to service provider 1. The secondary path is the T1 connection to service provider 2. AS path prepend is used to influence service provider 2 to direct inbound traffic destined to the public block over the Metro-Ethernet connection versus the T1 connection, except when the Metro-Ethernet connection is unavailable. This is accomplished with the route map (NOT-PREFERRED). This route map also automatically filters the routes that are advertised by matching only prefixes defined in the prefix list (PUBLIC-SUBNET) and dropping the rest due to the implicit *discard* all at the end of the route map.

Since the wide area network (WAN) connections are not of equal bandwidth, the customer also prefers to send outbound traffic over the Metro-Ethernet connection. A route map (PREFERRED) is used to create a preference for the default route learned from the neighbor across the Metro-Ethernet connection rather than the neighbor across the T1 connection. The route map is assigned to the Metro-Ethernet IPv4 eBGP neighbor (65.162.109.202) and matches a prefix list (DEFAULT) specifying the default route. The route map also applies a LOCAL_PREF value of 110 to the specified default route; making it more desirable than the default route learned from the T1 connection to service provider 2, which is assigned a default LOCAL_PREF value of 100.

When multihoming to two different service providers, it is good practice to advertise only intended networks to prevent becoming a transit AS. The customer's network in *Figure 6* could become a transit AS, if service provider 1 sent traffic destined for service provider 2 through the customer's AS (AS 500) or vice versa. This example uses an outbound prefix list (PUBLIC-SUBNET) to advertise only the customer public block to both service providers. This prefix list will prevent any routes learned by the AOS device using BGP from one service provider from being advertised to the other service provider. The prefix list is applied explicitly to the Metro-Ethernet AF neighbor with the **prefix-list PUBLIC-SUBNET out** command, and implicitly to the T1 neighbor through the NOT-PREFERRED route map applied outbound. If only default routes are learned from the service providers, the potential of becoming a transit AS is not an issue. However, it is good practice to use outbound prefixes as a preventative measure for multihoming setups.

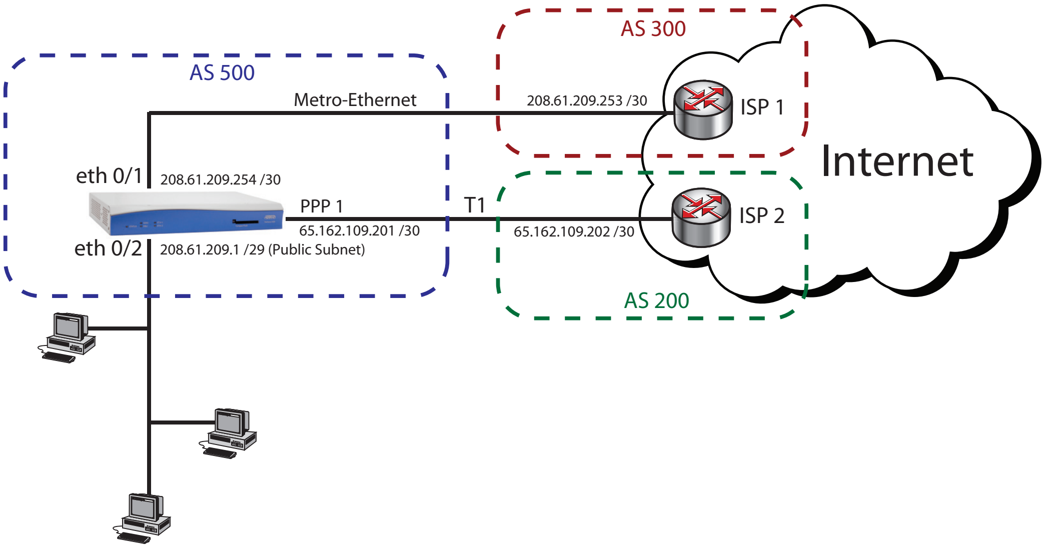


Figure 6. Multihoming to Two Different Service Providers

The following configuration applies to Example 3:

```
!
interface eth 0/1
    description 10 Mbps Metro-Ethernet connection to service provider 1
    ip address  208.61.209.254  255.255.255.252
    traffic-shape rate 10000000
    no shutdown
!
interface eth 0/2
    description Public Block of IPs being advertised to both service providers
    ip address  208.61.209.1  255.255.255.248
    no shutdown
!
```

```
interface t1 1/1
     clock source line
     tdm-group 1 timeslots 1-24 speed 64
     no shutdown
!
interface ppp 1
     description T1 connection to service provider 2
     ip address  65.162.109.201  255.255.255.252
     no shutdown
     cross-connect 1 t1 1/1 1 ppp 1
!
ip prefix-list PUBLIC-SUBNET seq 10 permit 208.61.209.0/29
ip prefix-list DEFAULT seq 10 permit 0.0.0.0/0
!
route-map NOT-PREFERRED permit 10
     match ip address prefix-list PUBLIC-SUBNET
     set as-path prepend 500 500 500 500 500
route-map PREFERRED permit 10
     match ip address prefix-list DEFAULT
     set local-preference 110
!
router bgp 500
     neighbor 65.162.109.202
          remote-as 200
          no shutdown
          exit
     neighbor 208.61.209.253
          remote-as 300
          no shutdown
          exit
     address-family ipv4
          network 208.61.209.0 mask 255.255.255.248
          neighbor 65.162.109.202
                no default-originate
                route-map NOT-PREFERRED out
                soft-reconfiguration inbound
               no shutdown
               exit
          neighbor 208.61.209.253
               prefix-list PUBLIC-SUBNET out
               route-map PREFERRED in
               soft-reconfiguration inbound
               exit
          exit
     exit
!
```

Consult with the service provider to determine which BGP attributes they will honor when making a decision on routing traffic back to your advertised AS.

Example 4: Load Sharing When Multihomed to Multiple Service Providers

AOS allows multiple equal cost routes to be used for the purposes of load sharing outbound traffic.

NOTE *The maximum number of equal cost routes supported in AOS is **6**.*

The need for load sharing is typically found in BGP applications where an AOS device is multihoming with multiple connections to different service providers. The BGP protocol does not provide support for load sharing. Therefore, BGP will always export the single best path for a given prefix to the IP route table. However, there are methods that can be implemented that will allow multiple BGP-derived routes to be imported into the IP route table. Aside from the BGP-specific configuration, load sharing must be globally enabled on the AOS device to allow the presence of multiple equal cost routes in the IP route table.

NOTE *At the global level, load sharing has two different implementation options: per packet and per destination. Refer to* Configuring IP Load Sharing in AOS *(ADTRAN's Knowledge Base article 1994) for more information on these load sharing options.*

The following example illustrates load sharing across multiple links where the customer's router is multihomed to two different service providers. Each service provider is advertising a default route to the AOS device. The default routes contain equal BGP attributes, therefore one route is no more desirable than the other according to the BGP selection process. The objective is to ensure that outbound traffic from the customer's network is load balanced (load shared) between the two Internet connections.

Several configuration steps are needed to allow BGP load sharing to take place. The **ip load-sharing per-destination** (for IPv4 routing) or the **ipv6 load-sharing per-destination** (for IPv6 routing) must be enabled in Global Configuration mode. This command allows duplicate routes to exist in the IPv4 or IPv6 routing table. The command **maximum-paths 2** is issued in BGP AF Configuration mode to allow up to two equal cost routes from BGP to be exported to the routing table. In order for multiple BGP routes to the same destination to be candidates for load sharing, they must be equal cost and share the same AS number in the AS path attribute. Since each service provider in this example owns their own unique AS number that is added on to the BGP path attribute, an arbitrary AS number must be prepended to each eBGP neighbor's BGP advertisement. This AS number enables the BGP routes to become candidates for load sharing since the routes now appear to originate from the same AS. This is accomplished with the route map (LOAD-SHARE) applied inbound to each eBGP neighbor. It matches the prefix list (DEFAULT) that only allows the default route, which automatically filters any other advertised routes. The route map then prepends the same arbitrary AS path (65009) to the default route prefix learned by each neighbor. The prefix list (PUBLIC-BLOCK) is used to advertise only the customer public block outbound to both service providers. This prefix will prevent any routes learned by the AOS device using BGP from one service provider from being advertised to the other service provider. If default routes are only learned from the service providers, then the potential of becoming a transit AS is not an issue. However, it is good practice to use outbound prefixes as a preventative measure for multihoming setups.

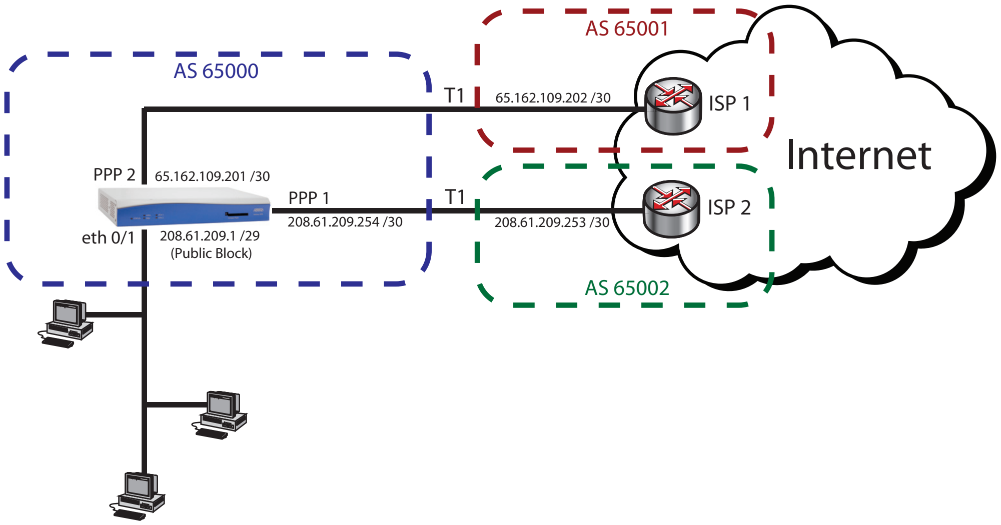


Figure 7. Load Balancing Across IPv4 Multihomed Connections to Two Service Providers

The following configuration applies to Example 4:

```
!
ip load-sharing per-destination
!
interface eth 0/1
    ip address  208.61.209.1 255.255.255.248
    no shutdown
!
interface t1 1/1
    clock source line
    tdm-group 1 timeslots 1-24 speed 64
    no shutdown
!
interface t1 2/1
    clock source line
    tdm-group 1 timeslots 1-24 speed 64
    no shutdown
!
interface ppp 1
    ip address  208.61.209.254  255.255.255.252
    no shutdown
    cross-connect 1 t1 1/1 1 ppp 1
!
interface ppp 2
    ip address  65.162.109.201  255.255.255.252
    no shutdown
    cross-connect 2 t1 2/1 1 ppp 2
!
ip prefix-list PUBLIC-BLOCK seq 10 permit 208.61.209.0/29
ip prefix-list DEFAULT seq 10 permit 0.0.0.0/0
```

```
!
route-map LOAD-SHARE permit 10
   match ip address prefix-list DEFAULT
   set as-path prepend 65009
!
router bgp 65000
   neighbor 65.162.109.202
      remote-as 65002
      no shutdown
      exit
   neighbor 208.61.209.253
      remote-as 65001
      no shutdown
      exit
   address-family ipv4
      maximum-paths 2
      network 208.61.209.0 mask 255.255.255.248
      neighbor 65.162.109.202
         prefix-list PUBLIC-BLOCK out
         route-map LOAD-SHARE in
         soft-reconfiguration inbound
         no shutdown
         exit
      neighbor 208.61.209.253
         prefix-list PUBLIC-BLOCK out
         route-map LOAD-SHARE in
         soft-reconfiguration inbound
         no shutdown
         exit
      exit
   exit
!
```

> *This configuration uses two separate T1 network interface modules (NIMs), allowing independent T1 clocking. The NetVanta dual T1 NIM (Part Number 1200872L1) does not allow independent clocking and cannot be used in applications where the T1 connections are terminated from two separate service providers.*

Example 5: Configuring Local Preference, MED, and Next-Hop-Self on an AOS Router with Both iBGP and eBGP Neighbors

The following example illustrates a scenario where an AOS router has both eBGP and iBGP IPv4 neighbors. The AOS device in AS 65000 has two eBGP neighbors: Remote Router 1 and Remote Router 2. This means that there are multiple exit points from the local AS (65000). One exit is through the AOS router over the Ethernet WAN interface (eth 0/1) to Remote Router 1. Another exit is through the AOS device over the PPP interface (PPP 1) to Remote Router 2.

The preferred path for traffic originating from the 192.168.1.0 /24 IPv4 network and destined for the remote private 172.16.5.0 /24 IPv4 network is the Metro-Ethernet connection. Since the remote network 172.16.5.0 /24 is advertised by both eBGP neighbors (Remote Routers 1 and 2), the local preference attribute is modified to ensure that the Metro-Ethernet connection is selected as the best path to the remote network. This modification is accomplished by creating a route map (SETLOCALPREF) that matches a prefix list (MATCHPREFIX), which specifies the 172.16.5.0 /24 IPv4 network. Within this route map, the LOCAL_PREF attribute is modified to 110, a higher value than the default LOCAL_PREF value of 100. The route map is applied inbound from the Ethernet WAN eBGP neighbor (192.168.2.2).

> NOTE *The route with the highest local preference value is preferred in BGP. Refer to BGP Path Selection on page 6 for more information about BGP preference rules.*

The Metro-Ethernet connection is also the preferred path for traffic originating from the remote 172.16.5.0 /24 IPv4 network inbound to the 192.168.1.0 /24 IPv4 network. The MED is modified to ensure that the Metro-Ethernet connection is selected as the best path inbound to the local network. This modification is accomplished by creating two separate route maps (SETMULTIEXIT1 and SETMULTIEXIT2); each matching a prefix list (NETWORK) that specifies the 192.168.1.0 /24 IPv4 network. The MULTI_EXIT_DISC value in the route map SETMULTIEXIT1 is set to 100, whereas the MULTI_EXIT_DISC value in the route map SETMULTIEXIT2 is set to 200. SETMULTIEXIT1 is applied outbound to the IPv4 Ethernet WAN eBGP neighbor (192.168.2.2). SETMULTIEXIT2 is applied outbound to the IPv4 T1 eBGP neighbor (10.10.10.2).

> NOTE *The route with the lowest MED metric value is preferred in BGP.*

The AOS device has two iBGP neighbors: Local Router 1 and Local Router 2. The **next-hop-self** command is used to advertise the IPv4 address of the AOS router (192.168.1.1) to its IPv4 iBGP neighbors as the path to reach any of the networks advertised by the eBGP peers.

> NOTE *All iBGP routers are fully meshed for both AS 65000 and AS 65001.*

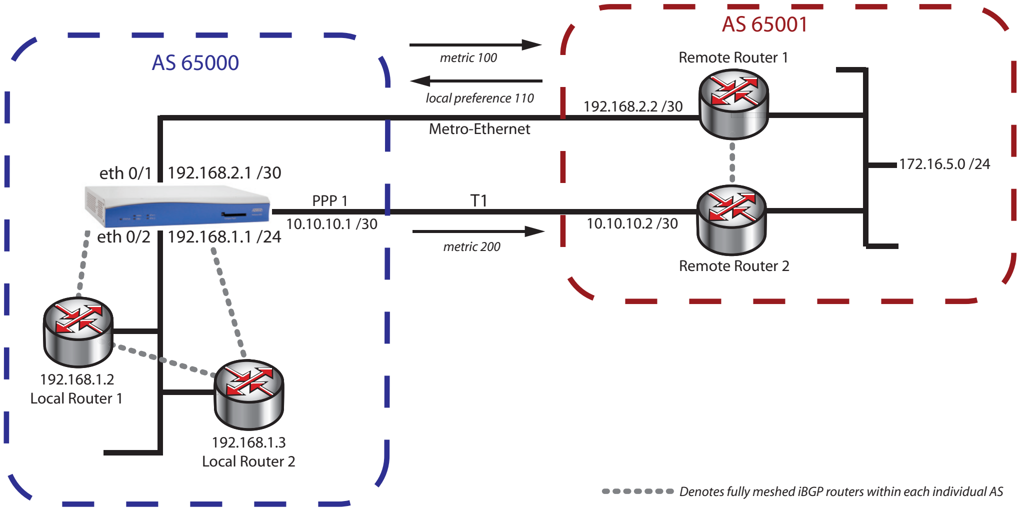


Figure 8. An AOS Device with Both iBGP and eBGP IPv4 Neighbors

The following configuration applies to Example 5:

```
    !
    interface eth 0/1
        ip address  192.168.2.1  255.255.255.252
        no shutdown
    !
    interface eth 0/2
        ip address  192.168.1.1  255.255.255.0
        no shutdown
    !
    interface t1 1/1
       clock source line
        tdm-group 1 timeslots 1-24 speed 64
        no shutdown
!
    interface ppp 1
        ip address  10.10.10.1  255.255.255.252
        no shutdown
        cross-connect 1 t1 1/1 1 ppp 1
    !
    ip prefix-list MATCHPREFIX seq 10 permit 172.16.5.0/24
    ip prefix-list NETWORK seq 10 permit 192.168.1.0/24
    !
    route-map SETLOCALPREF permit 10
        match ip address prefix-list MATCHPREFIX
        set local-preference 110
    route-map SETMULTIEXIT1 permit 10
        match ip address prefix-list NETWORK
        set metric 100
```

```
route-map SETMULTIEXIT2 permit 10
     match ip address prefix-list NETWORK
     set metric 200
!
router bgp 65000
    neighbor 192.168.2.2
        remote-as 65001
        no shutdown
        exit
    neighbor 10.10.10.2
        remote-as 65001
        no shutdown
        exit
    neighbor 192.168.1.2
        remote-as 65000
        no shutdown
        exit
    neighbor 192.168.1.3
        remote-as 65000
        no shutdown
        exit
    address-family ipv4
        network 192.168.1.0 mask 255.255.255.0
        neighbor 192.168.2.2
            no default-originate
            route-map SETLOCALPREF in
            route-map SETMULTIEXIT1 out
            soft-reconfiguration inbound
            no shutdown
            exit
         neighbor 10.10.10.2
             no default-originate
             route-map SETMULTIEXIT2 out
             soft-reconfiguration inbound
            no shutdown
            exit
        neighbor 192.168.1.2
             no default-originate
             next-hop-self
             soft-reconfiguration inbound
            no shutdown
            exit
        neighbor 192.168.1.3
             no default-originate
             next-hop-self
             soft-reconfiguration inbound
            no shutdown
            exit
        exit
!
```

Example 6: Using Local Preference to Promote a BGP Route as the Primary Internet Connection over a Backup Static Route

The following example illustrates how to configure an AOS device to prefer a default route learned from a service provider using eBGP as superior to a manually configured static route for the on-site backup Internet connection. In addition, the AOS device will advertise the default route to a local iBGP neighbor.

> *AOS only supports the **no default-originate** command, which prevents a unit from sending the default route to a BGP neighbor. However, transmission of default routes to BGP neighbors can be accomplished by issuing the command **network 0.0.0.0 mask 0.0.0.0**.*

An inbound prefix list (DEFAULT) is used in conjunction with a route map to ensure only the default route is learned from the service provider, and an outbound prefix list (FILTER) ensures no routes are advertised to the service provider from the AOS device. The DEFAULT prefix list is also used outbound to only allow the default route to be advertised to the iBGP neighbor.

In this example, an IPv4 default route is advertised from the service provider to the AOS device using BGP. A static default backup IPv4 route with an administrative distance of 30 is also specified in the AOS device (using the command **ip route 0.0.0.0 0.0.0.0 192.168.1.254 30**) for the on-site backup Internet connection. In addition, the **network 0.0.0.0 mask 0.0.0.0** command is issued to ensure that the AOS device transmits the default IPv4 route learned using BGP to its iBGP neighbor. Anytime a route is specified with the above network command or redistributed into BGP, it is subject to the BGP rules for determining the best path. Since the prefix of the static default backup route is specifically advertised by the network statement under BGP, the backup route is injected into the BGP process. This means that the default route advertised from the service provider and the static default backup route will be compared by BGP to determine the best path.

Both the advertised default route and the injected static route will be assigned the default LOCAL_PREF value of 100 in BGP. The learned default route advertised from the service provider is the preferred primary route. However, since both routes have the same value for LOCAL_PREF, the static default backup route will be selected as the best path by the BGP algorithm. The reason for this selection is because the static default backup route is a locally originated route, which is preferable to the BGP selection process than routes learned from a BGP neighbor. The eBGP learned route can be promoted as the preferred route by increasing its LOCAL_PREF value to 110. This is accomplished by applying an inbound route map (DEFAULT-ROUTE-IN) to the eBGP neighbor (service provider).

The route map match criteria specifies a prefix list (DEFAULT) that defines the default route advertised by the eBGP neighbor to the AOS device. When the match is made, the LOCAL_PREF value for that route is set to 110. Thus, the advertised default route is now more desirable with a LOCAL_PREF value of 110 than the static backup route with the default LOCAL_PREF value of 100. As a result, the eBGP learned default route is exported to the route table of the AOS device anytime the route is available. When the eBGP learned default route is not available, the AOS device will send traffic to the internal IPv4 backup Internet router (192.168.1.254) from the floating static default route.

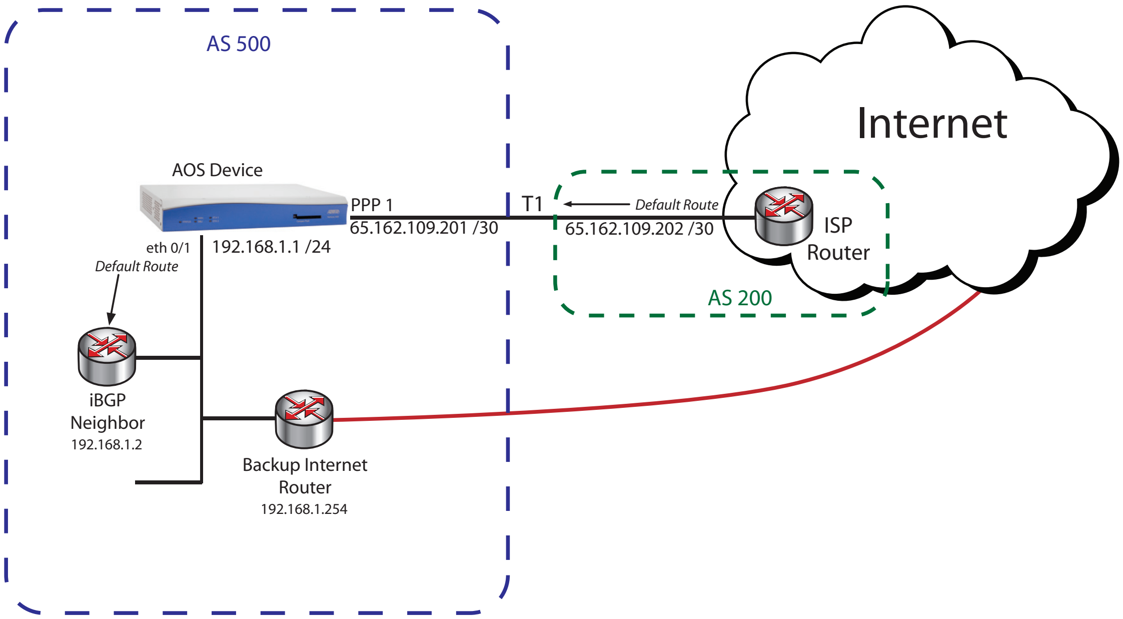


Figure 9. The BGP Advertised IPv4 Default Route Is the Preferred Primary Internet Connection

The following configuration applies to Example 6:

```
!
interface eth 0/1
    ip address  192.168.1.1  255.255.255.0
    no shutdown
!
interface t1 1/1
    clock source line
    tdm-group 1 timeslots 1-24 speed 64
    no shutdown
!
interface ppp 1
    ip address  65.162.109.201  255.255.255.252
    no shutdown
    cross-connect 1 t1 1/1 1 ppp 1
!
!
ip prefix-list DEFAULT seq 10 permit 0.0.0.0/0
ip prefix-list FILTER seq 10 deny 0.0.0.0/0 le 32
!
route-map DEFAULT-ROUTE-IN permit 10
    match ip address prefix-list DEFAULT
    set local-preference 110
!
router bgp 500
    neighbor 65.162.109.202
        remote-as 200
        no shutdown
        exit
    neighbor 192.168.1.2
        remote-as 500
        no shutdown
        exit
    address-family ipv4
        network 0.0.0.0 mask 0.0.0.0
        neighbor 65.162.109.202
            no default-originate
            prefix-list FILTER out
            route-map DEFAULT-ROUTE-IN in
            soft-reconfiguration inbound
            no shutdown
            exit
    neighbor 192.168.1.2
            no default-originate
            next-hop-self
            prefix-list DEFAULT out
            soft-reconfiguration inbound
            no shutdown
            exit
        exit
!
ip route 0.0.0.0 0.0.0.0 192.168.1.254 30
!
```

> *If the AOS router is used to provide Internet access to privately addressed hosts, many to one network address translation (NAT) needs to be configured by running the Firewall Wizard. Refer to Configuring Internet Access (Many to One NAT) with the Firewall Wizard in AOS (ADTRAN's Knowledge Base article 2185) for more information.*

Example 7: Using BGP Communities in an MPLS Network to Change Local Preference

The most common application for BGP community strings occurs in MPLS networks. Since MPLS providers tend to ignore the AS path attribute and the MED, a community string is often sent to signal the provider that the local preference should be changed. The change in local preference is necessary so that one prefix is seen as less preferable than another identical prefix. Consider the network in *Figure 10 on page 63*. There are three Internet-provisioned sites on an MPLS network (Primary, Secondary, and Tertiary) that will provide Internet access to remote sites on the network that do not have their own Internet circuits. All PE routers within the MPLS cloud are fully meshed iBGP neighbors. Routing information for the MPLS AS, including local preference information for exiting the AS, is synchronized among each of these PE routers.

The Primary site does not advertise a community string, and its prefixes are adopted with the default LOCAL_PREF value of 100. This local preference setting is the highest advertised in the MPLS network, making the Primary site the preferred connection for Internet traffic for the remote sites. A community string is used to manipulate the local preference in the MPLS cloud for the Secondary site, which will provide backup Internet access for the remote locations, if the Primary site's connection fails. A different community string is used to manipulate the local preference for the Tertiary site to make it the third backup

Internet connection. In this example, the MPLS provider has configured the community string 65000:90 for the Secondary site with a LOCAL_PREF value of 90. The community policy is defined using the command **set community 65000:90** under a route map (BGP-OUT) on the AOS device at the Secondary site. Similarly, the MPLS provider has configured the community string 65000:70 for the Tertiary site with a LOCAL_PREF value of 70. The community policy is defined using the command **set community 65000:70** under a route map (BGP-OUT) on the AOS device at the Tertiary site. The local preference attribute is shared and synchronized among all iBGP neighbors in the AS and serves to select the exit point out of the AS when multiple exit points exist for a particular route. In this example, the default route is chosen by local preference and triggers all of the PE routers to send customer Internet traffic to the designated Internet-provisioned customer site. Since the route with the largest local preference is preferred in BGP, the MPLS cloud will prefer the prefix with a LOCAL_PREF value of 100, 90, and finally 70, respectively.

NOTE *The local Internet connection at the Primary, Secondary, and Tertiary sites must be monitored by some device to determine when a failure occurs. Although this task can be performed by most AOS devices, this topic will not be covered in this configuration guide. For purposes of this example, the external firewall (see Figure 10 on page 63) will be monitoring the connection and advertising the default route to the MPLS router using RIP.*

The final configuration consideration is whether the Secondary and Tertiary sites prefer their own Internet connections to the Primary connection. Some organizations perform monitoring at the Primary site to keep track of their employees browsing habits, or to filter out certain sites from being available for browsing. In either case, the LOCAL_PREF value of the default route (learned from RIP in this example) must be set to either above or below the default LOCAL_PREF value in BGP (100) and redistributed into BGP. These settings allow the failover and failback Internet operations to function correctly. The Secondary site in this example prefers to use its own Internet connection rather than the Primary connection. So, the Secondary site's default prefix is redistributed into BGP and a route map (REDISTRIBUTE) is used to modify the LOCAL_PREF to the value 110 (a value greater than 100). The Tertiary site prefers to use the Primary site for Internet connectivity. Like the Secondary site, the Tertiary site's default prefix is also redistributed into BGP. However, the LOCAL_PREF value is set to 90 (a value less than 100).

NOTE *This example represents a typical MPLS BGP configuration where there is only one PE router that is an eBGP neighbor to the customer's router. Much of the BGP community configuration is done by the MPLS provider, leaving the customer with a much simpler configuration for the desired failover application. A sample configuration that shows what the configuration would look like if the MPLS cloud was a single AOS unit is provided at the end of this example.*

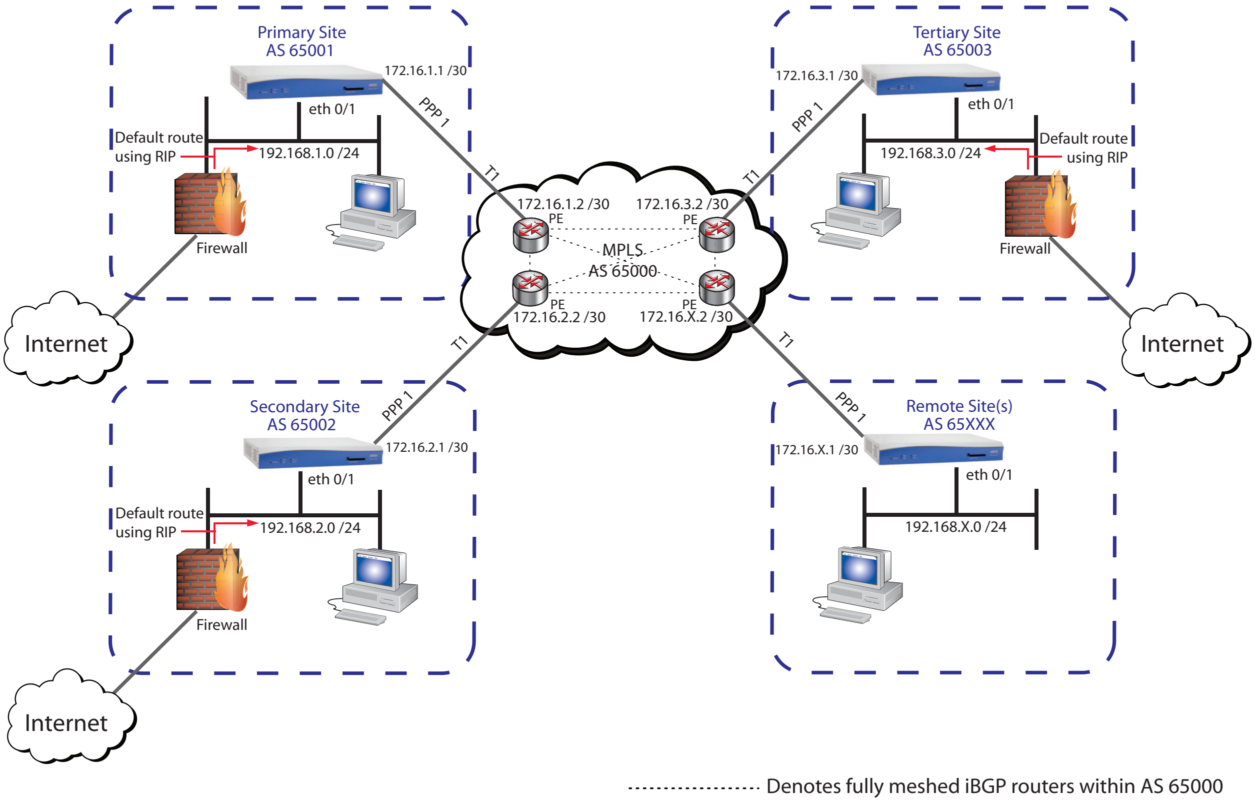


Figure 10. Three IPv4 Internet-Provisioned Sites on an MPLS Network Provide Internet Access for Remote Sites

The following configuration applies to Example 7:

Primary Site

In this example, the default route advertised with RIP by the external firewall (local prefix) is to be preferred.

```
!
interface eth 0/1
    ip address 192.168.1.1 255.255.255.0
    no shutdown
!
interface t1 1/1
    clock source line
    tdm-group 1 timeslots 1-24 speed 64
    no shutdown
!
interface ppp 1
    ip address 172.16.1.1 255.255.255.252
    no shutdown
    cross-connect 1 t1 1/1 1 ppp 1
```

```
!
ip prefix-list DEFAULT seq 10 permit 0.0.0.0/0
!
route-map REDISTRIBUTE permit 10
    match ip address prefix-list DEFAULT
    set local-preference 110
!
router bgp 65001
    neighbor 172.16.1.2
        remote-as 65000
        no shutdown
        exit
    address-family ipv4
        redistribute rip route-map REDISTRIBUTE
        network 192.168.1.0 mask 255.255.255.0
        neighbor 172.16.1.2
            no shutdown
            exit
        exit
!
```

Secondary Site

In this example, the default route advertised with RIP by the external firewall (local prefix) is to be preferred.

```
!
interface eth 0/1
    ip address 192.168.2.1 255.255.255.0
    no shutdown
!
interface t1 1/1
    clock source line
    tdm-group 1 timeslots 1-24 speed 64
    no shutdown
!
interface ppp 1
    ip address 172.16.2.1 255.255.255.252
    no shutdown
    cross-connect 1 t1 1/1 1 ppp 1
!
ip prefix-list DEFAULT seq 10 permit 0.0.0.0/0
!
route-map REDISTRIBUTE permit 10
    match ip address prefix-list DEFAULT
    set local preference 110
!
route-map BGP-OUT permit 10
    set community 65000:90
!
```

```
router bgp 65002
     neighbor 172.16.2.2
          remote-as 65000
          no shutdown
          exit
     address-family ipv4
          network 192.168.2.0 mask 255.255.255.0
          redistribute rip route-map REDISTRIBUTE
          neighbor 172.16.2.2
               route-map BGP-OUT out
               send-community standard
               no shutdown
               exit
          exit
!
```

Tertiary Site

In this example, the BGP prefix is to be preferred.

```
!
interface eth 0/1
     ip address 192.168.3.1 255.255.255.0
     no shutdown
!
interface t1 1/1
     clock source line
     tdm-group 1 timeslots 1-24 speed 64
     no shutdown
!
interface ppp 1
     ip address 172.16.3.1 255.255.255.252
     no shutdown
     cross-connect 1 t1 1/1 1 ppp 1
!
ip prefix-list DEFAULT seq 10 permit 0.0.0.0/0
!
route-map REDISTRIBUTE permit 10
     match ip address prefix-list DEFAULT
     set local-preference 90
!
route-map BGP-OUT permit 10
     set community 65000:70
!
router bgp 65003
     neighbor 172.16.3.2
          remote-as 65000
          no shutdown
          exit
     address-family ipv4
          redistribute rip route-map REDISTRIBUTE
```

```
            network 192.168.3.0 mask 255.255.255.0
            neighbor 172.16.3.2
                route-map BGP-OUT out
                send-community standard
                no shutdown
                exit
            exit
    !
```

Remote Site(s)

In this example, there is no local Internet connection, so it requires no special configuration. The Internet access for this site would come from the MPLS cloud using the site currently chosen by local preference.

```
!
interface eth 0/1
    ip address 192.168.X.1 255.255.255.0
    no shutdown
!
interface t1 1/1
    clock source line
    tdm-group 1 timeslots 1-24 speed 64
    no shutdown
!
interface ppp 1
    ip address 172.16.X.1 255.255.255.252
    no shutdown
    cross-connect 1 t1 1/1 1 ppp 1
!
router bgp 65XXX
    neighbor 172.16.X.2
        remote-as 65000
        no shutdown
        exit
    address-family ipv4
        network 192.168.X.0 mask 255.255.255.0
        neighbor 172.16.X.2
            soft reconfiguration inbound
            no shutdown
            exit
        exit
!
```

MPLS Cloud Sample Configuration

> NOTE *The following configuration represents the BGP community setup and full meshing in the MPLS cloud* ***if*** *it were represented on a single AOS device. The provided configuration is only intended to be an example for administrators whose network is performing the full meshing and BGP definitions that is normally realized by the MPLS network.*

```
!
community-list C100
    permit 65000:100
!
community-list C90
    permit 65000:90
!
community-list C70
    permit 65000:70
!
route-map BGP-IN permit 10
    match community C100
    set local-preference 100
!
route-map BGP-IN permit 20
    match community C90
    set local-preference 90
!
route-map BGP-IN permit 30
    match community C70
    set local-preference 70
!
route-map BGP-IN permit 40
!
router bgp 65000
    neighbor 172.16.1.1
        remote-as 65001
        no shutdown
        exit
    neighbor 172.16.2.1
        remote-as 65002
        no shutdown
        exit
    neighbor 172.16.3.1
        remote-as 65003
        no shutdown
        exit
    neighbor 172.16.X.1
        remote-as 65XXX
        no shutdown
        exit
    address-family ipv4
        neighbor 172.16.1.1
    route-map BGP-IN in
    send-community standard
    no shutdown
    exit
neighbor 172.16.2.1
    route-map BGP-IN in
    send-community standard
    no shutdown
    exit
neighbor 172.16.3.1
    route-map BGP-IN in
    send-community standard
    no shutdown
    exit
neighbor 172.16.X.1
    route-map BGP-IN in
    send-community standard
    no shutdown
    exit
exit
```

Example 8: Configuring BGP Using Multiple VRFs

In this example, multiple VRF instances are used to connect multiple locations of a corporation. Each branch connected to the AOS CE router, branches 1, 2, and 3, has its own VRF configured on the AOS router. Branch 1 operates on VRF RED, Branch 2 operates on VRF BLUE, and Branch 3 operates on VRF GREEN. Each of these branches needs to communicate and share data with corporate branches on the provider's side of the MPLS network. Therefore, the AOS device is configured with BGP in order to facilitate communication between the various branches by maintaining the VRF divisions from one side of the AOS router to the other side of the MPLS network. BGP transports the VRFs (RED, BLUE, and GREEN) across a subinterface link between the AOS device and the provider's router (PE). The PE network maintains the VRF divisions, and connects correctly with the appropriate corporate branches on the provider's MPLS network (Branch 4 using VRF GREEN, Branch 5 using VRF RED, and Branch 6 using VRF BLUE).

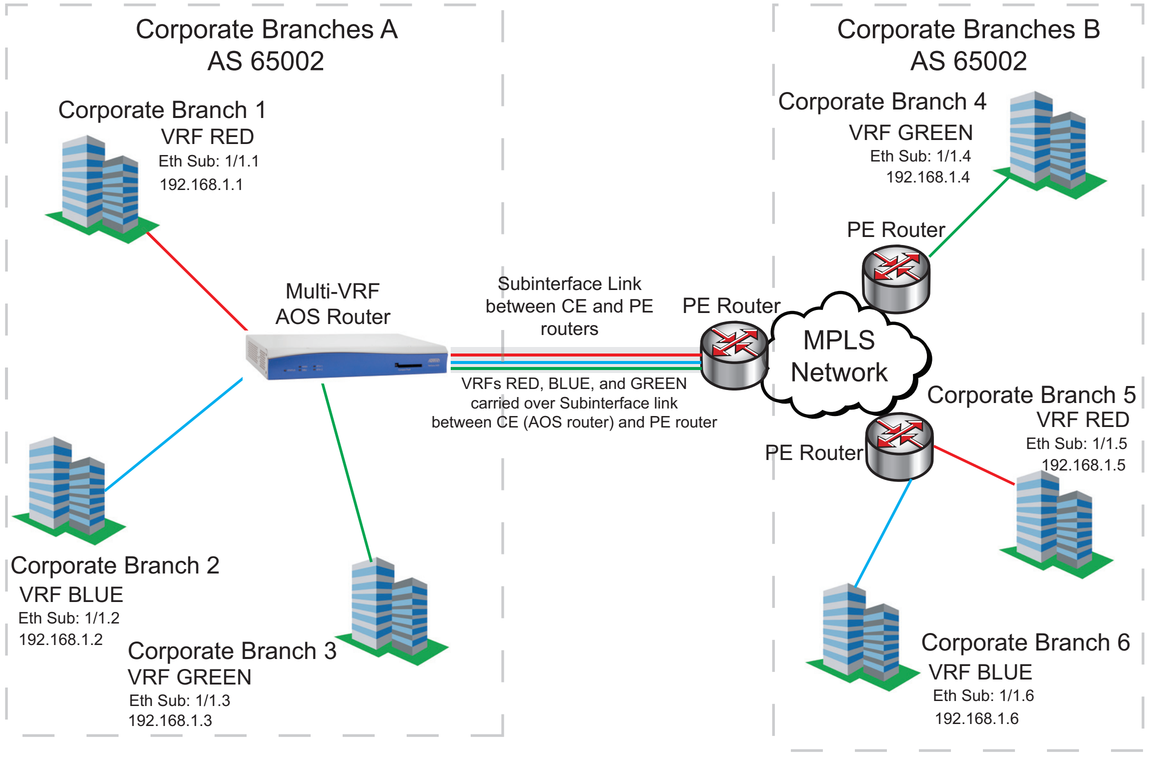


Figure 11. IPv4 Multi-VRF BGP Configuration

The basics of this configuration include configuring the three VRFs, configuring the BGP session for each VRF on the appropriate subinterface (Ethernet in this case), and creating the IPv4 BGP neighbors for each VRF. The following configuration applies to Example 8:

Corporate Branches A

```
!
interface eth 1/1.1
    ip address 192.168.1.1 255.255.255.0
    no shutdown
!
interface eth 1/1.2
    ip address 192.168.1.2 255.255.255.0
    no shutdown
!
interface eth 1/1.3
    ip address 192.168.1.3 255.255.255.0
    no shutdown
!
vrf RED route-distinguisher as-2byte 65002:11
vrf BLUE route-distinguisher as-2byte 65002:222
vrf GREEN route-distinguisher as-2byte 65002:333
!
router bgp 65002
```

```
 vrf RED
  neighbor 192.168.1.5
   remote-as 65002
   no shutdown
   exit
  address-family ipv4
   network 192.168.1.0 mask 255.255.255.0
   neighbor 192.168.1.5
    soft-reconfiguration inbound
    no shutdown
    exit
   exit
  exit
 vrf BLUE
  neighbor 192.168.1.6
   remote-as 65002
   no shutdown
   exit
  address-family ipv4
   network 192.168.1.0 mask 255.255.255.0
   neighbor 192.168.1.6
    soft-reconfiguration inbound
    no shutdown
    exit
   exit
  exit
 vrf GREEN
  neighbor 192.168.1.4
   remote-as 65002
   no shutdown
   exit
  address-family ipv4
   network 192.168.1.0 mask 255.255.255.0
   neighbor 192.168.1.4
    soft-reconfiguration inbound
    no shutdown
    exit
   exit
  exit
!
```

Corporate Branches B

```
!
interface eth 1/1.4
 ip address 192.168.1.4 255.255.255.255
 no shutdown
!
```

```
interface eth 1/1.5
    ip address 192.168.1.5 255.255.255.255
    no shutdown
!
interface eth 1/1.6
    ip address 192.168.1.6 255.255.255.255
    no shutdown
!
vrf RED route-distinguisher as-2byte 65002:111
vrf BLUE route-distinguisher as-2byte 65002:222
vrf GREEN route-distinguisher as-2byte 65002:333
!
router bgp 65002
    vrf RED
        neighbor 192.168.1.1
            remote-as 65002
            no shutdown
            exit
        address-family ipv4
            network 192.168.1.0 mask 255.255.255.0
            neighbor 192.168.1.1
                soft-reconfiguration inbound
                no shutdown
                exit
            exit
        exit
    vrf BLUE
        neighbor 192.168.1.2
            remote-as 65002
            no shutdown
            exit
        address-family ipv4
            network 192.168.1.0 mask 255.255.255.0
            neighbor 192.168.1.2
                soft-reconfiguration inbound
                no shutdown
                exit
            exit
        exit
    vrf GREEN
        neighbor 192.168.1.3
            remote-as 65002
            no shutdown
            exit
        address-family ipv4
            network 192.168.1.0 mask 255.255.255.0
            neighbor 192.168.1.3
                soft-reconfiguration inbound
                no shutdown
```

```
                   exit
              exit
         exit
!
```

BGP Configuration Command Summary

The following tables summarize the configuration commands associated with BGP.

It is important to note that BGP sessions must be cleared for BGP policy changes, such as alterations to the prefix list filters, to take effect. Use the ***clear bgp*** *command to clear BGP neighbors. Typically, soft resets should be used because hard resets can disrupt the network. Refer to Clear BGP on page 100 for detailed information on how to properly use this command.*

Table 3. Basic BGP Configuration Steps

	Prompt	Command	Description
Step 1	(config)#	**router bgp** *<AS number>*	Enables BGP and specifies the local AS. Range is **1** to **4294967295**.
Step 2	(config-bgp)#	**neighbor [***<ipv4 address>* | *<ipv6 address>***]**	Specifies the IPv4 or IPv6 address for the BGP neighbor on the default VRF.
Step 3	(config-bgp-neighbor)#	**remote-as** *<value>*	Specifies the AS to which this neighbor on the default VRF belongs. Range is **1** to **4294967295**.
Step 4	(config-bgp-neighbor)#	**no shutdown**	Activates the BGP neighbor on the default VRF.
Step 5	(config-bgp-neighbor)#	**exit**	Exits the BGP Neighbor Configuration mode, and returns to the BGP Configuration mode.
Step 6	(config-bgp)#	**address-family [ipv4 | ipv6]**	Creates the AF on the default VRF and enters the AF's configuration mode.
Step 7	(config-bgp-ipv4)# (config-bgp-ipv6)#	**network** *<ipv4 address>* **mask** *<subnet mask>* OR **network** *<ipv6 address/prefix-length>*	Specifies the local networks that remote sites should be able to access for this AF on the default VRF.
Step 8	(config-bgp-ipv4)# (config-bgp-ipv6)#	**neighbor [***<ipv4 address>* | *<ipv6 address>***]**	Specifies the AF neighbor on the default VRF.
Step 9	(config-bgp-ipv4-neighbor)# (config-bgp-ipv6-neighbor)#	**no shutdown**	Activates the AF neighbor on the default VRF.

Table 3. Basic BGP Configuration Steps

	Prompt	Command	Description
Step 10	(config-bgp-ipv4-neighbor)# (config-bgp-ipv6-neighbor)#	**exit**	Exits the BGP AF Neighbor Configuration mode, and returns to the BGP AF Configuration mode.
Step 11	(config-bgp-ipv4)# (config-bgp-ipv6)#	**exit**	Exits the BGP AF Configuration mode, and returns to BGP Configuration mode.
Step 12	(config-bgp)#	**vrf** *<name>*	Specifies a nondefault VRF and enters the BGP nondefault VRF Configuration mode.
Step 13	(config-bgp-vrf)#	**neighbor [***<ipv4 address>* | *<ipv6 address>***]**	Specifies the IPv4 or IPv6 address for the BGP neighbor on the nondefault (named) VRF.
Step 14	(config-bgp-vrf-neighbor)#	**no shutdown**	Activates the BGP neighbor on the nondefault (named) VRF.
Step 15	(config-bgp-vrf-neighbor)#	**remote-as** *<value>*	Specifies the AS to which this neighbor on the nondefault (named) VRF belongs. Range is **1** to **4294967295**.
Step 16	(config-bgp-vrf-neighbor)#	**exit**	Exits the BGP VRF Neighbor Configuration mode, and returns to the BGP VRF Configuration mode.
Step 17	(config-bgp-vrf)#	**address-family [ipv4 | ipv6]**	Creates the AF on the nondefault VRF and enters the AF's configuration mode.
Step 18	(config-bgp-vrf-ipv4)# (config-bgp-vrf-ipv6)#	**network** *<ipv4 address>* **mask** *<subnet mask>* OR **network** *<ipv6 address/prefix-length>*	Specifies the local networks that remote sites should be able to access for this AF on the nondefault (named) VRF.
Step 19	(config-bgp-vrf-ipv4)# (config-bgp-vrf-ipv6)#	**neighbor [***<ipv4 address>* | *<ipv6 address>***]**	Specifies the AF neighbor on the nondefault (named) VRF.
Step 20	(config-bgp-vrf-ipv4-neighbor)# (config-bgp-vrf-ipv6-neighbor)#	**no shutdown**	Activates the AF neighbor on the nondefault (named) VRF.
Step 21	(config-bgp-vrf-ipv4-neighbor)# (config-bgp-vrf-ipv6-neighbor)#	**exit**	Exits the BGP AF Neighbor Configuration mode on the nondefault VRF, and returns to BGP AF Neighbor Configuration mode on the default VRF.
Step 22	(config-bgp-vrf-ipv4)# (config-bgp-vrf-ipv6)#	**do write**	Saves the configuration.

Refer to Basic BGP Configuration Using the CLI on page 11 for detailed information on the commands used in the table above.

Table 4. Additional BGP Global Configuration Options

Prompt	Command	Description
(config-bgp)#	**bgp fast-external-failover**	Enables the fast external failover feature.
(config-bgp)#	**hold-timer** *<value>*	Specifies a default hold time for all neighbors in this BGP process. This is the time interval (in seconds) within which a keepalive must be received from a peer before that peer is declared dead. Range is **0** to **65535** seconds.
(config-bgp)#	**bgp log-neighbor-changes**	Turns on the logging of BGP neighbor state changes. State change messages will appear on the screen when connected to the console port.
(config-bgp)#	**bgp [always-compare-med | compare-med | deterministic-med | ignore-med]**	Instructs AOS on how to handle MEDs for all routes from the same AS.
(config-bgp)#	**bgp router-id** *<ipv4 address>*	Specifies the router ID that the BGP interface uses to identify itself to its neighbors. IPv4 addresses should be expressed in dotted decimal notation (for example, **10.10.10.1**).

Table 5. Additional BGP Neighbor Configuration Options

Prompt	Command	Description
(config-bgp-neighbor)# (config-bgp-vrf-neighbor)#	**description** *<text>*	Identifies the specified interface connected to the BGP neighbor (on either the default VRF or nondefault (named) VRF) using up to **80** alphanumeric characters.
(config-bgp-neighbor)# (config-bgp-vrf-neighbor)#	**ebgp-multihop** *<value>*	Configures the maximum hop count of BGP messages to a neighbor on either the default VRF or nondefault (named) VRF. The default TTL for BGP messages is **1**. Range is **1** to **255** hops.
(config-bgp-neighbor)# (config-bgp-vrf-neighbor)#	**hold-timer** *<value>*	Specifies a default hold time for this BGP neighbor on either the default VRF or nondefault (named) VRF. This is the time interval (in seconds) within which a keepalive must be received from the peer before that peer is declared dead. Range is 0 to **65535** seconds.
(config-bgp-neighbor)# (config-bgp-vrf-neighbor)#	**local-as** *<value>*	Specifies an AS number that is different from the one specified in the **router bgp** command. The AS number is used when communicating with this BGP neighbor on either the default VRF or nondefault (named) VRF.
(config-bgp-neighbor)# (config-bgp-vrf-neighbor)#	**password** *<password>*	Enables MD5 password authentication to a BGP peer and specifies the password string to be used for authentication. This setting is applicable to BGP neighbors on either the default VRF or nondefault (named) VRF.
(config-bgp-neighbor)# (config-bgp-vrf-neighbor)#	**transport connection-mode [active | passive]**	Specifies that only active or passive TCP session connections are allowed for the neighbor on either the default VRF or nondefault (named) VRF.
(config-bgp-neighbor)# (config-bgp-vrf-neighbor)#	**update-source** *<interface>*	Specifies which interface's IP address will be used as the source IP address for the BGP TCP connection. This setting is applicable to BGP neighbors on either the default VRF or nondefault (named) VRF, however, the interface must exist in the same VRF as the neighbor being configured. If the interface is in a different VRF, this setting is ignored.

NOTE

Refer to Additional BGP Neighbor Settings (for both default VRF and nondefault VRF) on page 17 for more detailed information on the commands referenced in the table above.

Table 6. Additional BGP AF Configuration Options

Prompt	Command	Description
(config-bgp-ipv4)# (config-bgp-vrf-ipv4)# (config-bgp-ipv6)# (config-bgp-vrf-ipv6)#	**distance bgp** *<external>* *<internal>* *<local>*	Specifies the administrative distance for BGP routes learned via eBGP (external), iBGP (internal), the **network** command and redistributed routes (local). This setting applies to the AF on either the default VRF or nondefault (named) VRF. By default, external is set to **20**, internal to **200**, and local to **200**.
(config-bgp-ipv4)# (config-bgp-vrf-ipv4)# (config-bgp-ipv6)# (config-bgp-vrf-ipv6)#	**redistribute [connected | ospf | rip | static] [metric** *<value>* **| route-map** *<map>***]**	Used to make a static route or a prefix learned from another routing protocol more or less preferable than any route learned using BGP. This command applies to an AF on either the default VRF or nondefault (named) VRF. Establishing routing preferences in this manner is discussed in *Establishing Routing Preference on page 24*.
(config-bgp-ipv4)# (config-bgp-vrf-ipv4)# (config-bgp-ipv6)# (config-bgp-vrf-ipv6)#	**maximum-paths** *<value>*	Specifies the maximum number of equal cost parallel routes (shared paths) BGP can export to the route table. This command applies to an AF on either the default VRF or nondefault (named) VRF. Range is **1** to **6**. For proper configuration, set this value to **2** or more.

Table 7. Additional BGP AF Neighbor Configuration Options

Prompt	Command	Description
(config-bgp-ipv4-neighbor)# (config-bgp-vrf-ipv4-neighbor)# (config-bgp-ipv6-neighbor)# (config-bgp-vrf-ipv6-neighbor)#	**advertisement-interval** *<value>*	Specifies the minimum interval between sending updates to the neighbor. The default advertisement interval is **30** seconds for external neighbors and **5** seconds for internal neighbors. Range is **0** to **600** seconds. This command applies to AF neighbors on the default VRF or nondefault (named) VRF.
(config-bgp-ipv4-neighbor)# (config-bgp-vrf-ipv4-neighbor)# (config-bgp-ipv6-neighbor)# (config-bgp-vrf-ipv6-neighbor)#	**as-path-list** *<name>* **[in | out]**	Specifies the name of the AS path list that will be used to filter routes on the AF neighbor. The optional **in** and **out** keywords determine whether the filter is applied to inbound or outbound traffic, respectively. This command applies to AF neighbors on the default VRF or nondefault (named) VRF.

Table 7. Additional BGP AF Neighbor Configuration Options *(Continued)*

Prompt	Command	Description
(config-bgp-ipv4-neighbor)# (config-bgp-vrf-ipv4-neighbor)# (config-bgp-ipv6-neighbor)# (config-bgp-vrf-ipv6-neighbor)#	**send-community standard**	Enables this AOS device to accept a community attribute or adds a community attribute to any advertisement sent by this peer. This command applies to AF neighbors on the default VRF or nondefault (named) VRF.
(config-bgp-ipv4-neighbor)# (config-bgp-vrf-ipv4-neighbor)# (config-bgp-ipv6-neighbor)# (config-bgp-vrf-ipv6-neighbor)#	**distribute-list** *<ipv4/ipv6 acl name>* **[in | out]**	Assigns an inbound or outbound IPv4 or IPv6 ACL to this BGP neighbor for filtering. This command applies to AF neighbors on the default VRF or nondefault (named) VRF.
(config-bgp-ipv4-neighbor)# (config-bgp-vrf-ipv4-neighbor)# (config-bgp-ipv6-neighbor)# (config-bgp-vrf-ipv6-neighbor)#	**next-hop-self**	Forces the next-hop attribute to be changed to this unit's address for each network it advertises to the BGP neighbor. This command applies to AF neighbors on the default VRF or nondefault (named) VRF.
(config-bgp-ipv4-neighbor)# (config-bgp-vrf-ipv4-neighbor)# (config-bgp-ipv6-neighbor)# (config-bgp-vrf-ipv6-neighbor)#	**no default-originate**	Prevents the unit from sending the default route to a BGP AF neighbor. This is a default setting and cannot be modified. A default route can be transmitted to a BGP neighbor by manually entering a default route in BGP Configuration mode with the **network 0.0.0.0 mask 0.0.0.0** command. This command applies to AF neighbors on the default VRF or nondefault (named) VRF.
(config-bgp-ipv4-neighbor)# (config-bgp-vrf-ipv4-neighbor)# (config-bgp-ipv6-neighbor)# (config-bgp-vrf-ipv6-neighbor)#	**prefix-list** *<name>* **[in | out]**	Assigns a prefix list to a BGP AF neighbor and specifies whether the list will be used to filter inbound or outbound routes. This command applies to AF neighbors on the default VRF or nondefault (named) VRF.

Table 7. Additional BGP AF Neighbor Configuration Options ***(Continued)***

Prompt	Command	Description
(config-bgp-ipv4-neighbor)# (config-bgp-vrf-ipv4-neighbor)# (config-bgp-ipv6-neighbor)# (config-bgp-vrf-ipv6-neighbor)#	**route-map** <*name*> **[in | out]**	Applies a previously configured route map entry to a BGP neighbor and specifies whether this route map will filter/modify inbound or outbound BGP route updates. This command applies to AF neighbors on the default VRF or nondefault (named) VRF.
(config-bgp-ipv4-neighbor)# (config-bgp-vrf-ipv4-neighbor)# (config-bgp-ipv6-neighbor)# (config-bgp-vrf-ipv6-neighbor)#	**soft-reconfiguration inbound**	Enables the AOS device to store BGP updates for the specified neighbor. This command applies to AF neighbors on the default VRF or nondefault (named) VRF.

Table 8. Filtering Routes and Applying Attributes with Route Maps

	Prompt	Command	Description
Step 1	(config)#	**route-map** <*name*> **[deny | permit]** <*number*>	Creates a route map and enters the Route Map Configuration mode. Range is **1** to **4294967295**.
Step 2 *(see note)*	Select routes for the BGP interface to advertise outbound or filter inbound using the **match** commands.		
	(config-route-map)#	**match ip address prefix-list** <*name*> OR **match ipv6 address prefix-list**	Filters according to a prefix list.
	(config-route-map)#	**match ip address** <*ipv4 acl name*> OR **match ipv6 address** <*ipv6 acl name*>	Filters according to an IPv4 or IPv6 ACL.
	(config-route-map)#	**match as-path** <*name*>	Filters according to AS path.
	(config-route-map)#	**match community** <*name*> **[exact-match]**	Filters according to a community list.
	(config-route-map)#	**match metric** <*value*>	Filters according to metric.

Table 8. Filtering Routes and Applying Attributes with Route Maps *(Continued)*

	Prompt	Command	Description
Step 3 (Optional)	Apply attributes to the route using the **set** commands.		
	(config-route-map)#	**set as-path prepend [**<*number*> **| last-as** <*number*>**]**	Prepends AS hops to the selected route(s). The **last-as** keyword is optional and specifies to repeat the last AS in a route up to **10** times. If the **last-as** keyword is not used, you can enter multiple AS numbers to be prepended to the UPDATE message.
	(config-route-map)#	**set metric** <*value*>	Specifies a MED value. Range is **0** to **4294967295**.
	(config-route-map)#	**set local-preference** <*value*>	Specifies a local preference value. Range is **0** to **4294967295**.
Step 4	(config-route-map)#	**exit**	Exits Route Map Configuration mode and returns to the Global Configuration mode.
Step 5	(config)#	**router bgp** <*AS number*>	Enters BGP Configuration mode. Range is **1** to **4294967295**.

Table 8. Filtering Routes and Applying Attributes with Route Maps ***(Continued)***

	Prompt	Command	Description
Step 6	(config-bgp)#	**address-family [ipv4 | ipv6]**	Enters BGP AF Configuration mode for the neighbor to which the route map is to be applied.
Step 7	(config-bgp-ipv4)# (config-bgp-ipv6)#	**neighbor [**<*ipv4 address*> | <*ipv6 address*>**]**	Enters BGP AF Neighbor Configuration mode for the neighbor to which the route map is to be applied.
Step 8	(config-bgp-ipv4-neighbor)# (config-bgp-vrf-ipv4-neighbor)# (config-bgp-ipv6-neighbor)# (config-bgp-vrf-ipv6-neighbor)#	**route-map** <*name*> **[in | out]**	Applies the route map to a BGP neighbor and specifies whether it is to filter/modify inbound or outbound routes.
Step 9	(config-bgp-ipv4-neighbor)# (config-bgp-vrf-ipv4-neighbor)# (config-bgp-ipv6-neighbor)# (config-bgp-vrf-ipv6-neighbor)#	**do write**	Saves the configuration.

NOTE *Lists referenced in Step 2 (prefix, ACL, AS path, and community) first must be created separately before they can be used within the route map for filtering. Refer to Route Maps on page 37 for additional information on route maps and related commands used to filter routes and apply attributes.*

Table 9. Defining a Community Policy

	Prompt	Command	Description
Step 1	(config)#	**community-list** *<name>*	Creates a community list for a BGP route map to use. The communities defined in this command should match an existing community string that the AOS device is receiving.
Step 2	(config-comm-list)#	**[permit | deny] [***<value>* **| internet | local-as | no-advertise | no-export]**	Adds an entry to the community list that either permits or denies BGP routes containing the specified community string in the community attribute.
	(config-comm-list)#	**exit**	Exits the Community List Configuration mode and returns to the Global Configuration mode.
Step 3	(config)#	**route-map** *<name>* **[deny | permit]** *<number>*	Creates a route map, if one does not already exist, to reference the community list.
Step 4	(config-route-map)#	**match community** *<name>* **[exact-match]**	References the *<name>* of the community list created in Step 1 to add it to the route map entry. The **exact-match** keyword is optional and specifies that the route map must match all configured community strings in the specified list exactly.

Table 9. Defining a Community Policy *(Continued)*

	Prompt	Command	Description
Step 5	Use **set** commands to configure any policies to be applied to the community.		
	(config-route-map)#	**set as-path prepend** <*number*> **[last-as]** <*number*>	Prepends AS hops to the selected route(s). The **last-as** keyword is optional and specifies to repeat the last AS in a route up to **10** times.
	(config-route-map)#	**set metric** <*value*>	Specifies a MULTI_EXIT_DISC value for the selected route(s). Range is **0** to **4294967295**.
	(config-route-map)#	**set local-preference** <*value*>	Changes the LOCAL_PREF value for selected route(s). Range is **0** to **4294967295**.
Step 6	(config-route-map)#	**exit**	Exits Route Map Configuration mode and returns to the Global Configuration mode.
Step 7	(config)#	**router bgp** <*AS number*>	Enters BGP Configuration mode. Range is **1** to **4294967295**.
Step 8	(config-bgp)#	**address-family [ipv4 | ipv6]**	Enters BGP AF Configuration mode.
Step 9	(config-bgp-ipv4)# (config-bgp-ipv6)#	**neighbor [**<*ipv4 address*> | <*ipv6 address*>**]**	Enters BGP AF Neighbor Configuration mode for the neighbor to which the route map is to be applied.
Step 10	(config-bgp-ipv4-neighbor)# (config-bgp-ipv6-neighbor)#	**send-community standard**	Enables this AOS device to accept a community attribute or adds a community attribute to any advertisement sent by this peer.
Step 11	(config-bgp-ipv4-neighbor)# (config-bgp-ipv6-neighbor)#	**route-map** <*name*> **in**	Applies the route map to the BGP neighbor as an inbound policy.
Step 12	(config-bgp-ipv4-neighbor)# (config-bgp-ipv6-neighbor)#	**do write**	Saves the configuration.

NOTE *Refer to AS-Path List on page 28 for additional configuration options and more detailed information on the BGP community commands listed in Table 9 on page 81.*

Table 10. Advertising a BGP Community

	Prompt	Command	Description
Step 1	(config)#	**ip prefix-list** *<name>* **seq** *<number>* **[deny | permit]** *<network ip/length>* **[ge | le]** *<value>* OR **ipv6 prefix-list** *<name>* **seq** *<number>* **[deny | permit]** *<ipv6 address/prefix-length>* **[ge | le]** *<value>*	Creates a prefix list to define the routes that are to be tagged with a community string. Range is **1** to **4294967294**.
Step 2	(config)#	**route-map** *<name>* **[deny | permit]** *<number>*	Creates a route map (if one does not already exist) to reference the prefix list. Range is **1** to **4294967294**.
Step 3	(config-route-map)#	**match ip address prefix-list** *<name>* OR **match ipv6 address prefix-list** *<name>*	References the *<name>* of the prefix list created in Step 1 to add it to the route map entry.
Step 4	(config-route-map)#	**set community** *<value>* **[add | internet | local-as | no-advertise | no-export]**	Sets the community attribute to either a privately defined community or one of the well-known community numbers for routes serviced by this route map.
Step 5	(config-route-map)#	**exit**	Exits Route Map Configuration mode and returns to the Global Configuration mode.
Step 6	(config)#	**router bgp** *<AS number>*	Enters BGP Configuration mode. Range is **1** to **4294967295**.
Step 7	(config-bgp)#	**address-family [ipv4 | ipv6]**	Enters BGP AF Configuration mode.

Table 10. Advertising a BGP Community *(Continued)*

	Prompt	Command	Description
Step 8	(config-bgp-ipv4)# (config-bgp-ipv6)#	**neighbor [***<ipv4 address>* **|** *<ipv6 address>***]**	Enters BGP AF Neighbor Configuration mode for the neighbor to which the route map is to be applied.
Step 9	(config-bgp-ipv4-neighbor)# (config-bgp-ipv6-neighbor)#	**send-community standard**	Enables this AOS device to accept a community attribute or add a community attribute to any advertisement sent by this peer.
Step 10	(config-bgp-ipv4-neighbor)# (config-bgp-ipv6-neighbor)#	**route-map** *<name>* **out**	Applies the route map to the BGP neighbor as an outbound policy.
Step 11	(config-bgp-ipv4-neighbor)# (config-bgp-ipv4-neighbor)#	**do write**	Saves the configuration.

NOTE *Refer to Prefix List on page 35 for more information on creating prefix lists. Refer to Step 2: Defining a Community Policy on page 31 for more information on the* ***set community*** *command.*

Table 11. Deleting a Community

	Prompt	Command	Description
Step 1	(config)#	**ip prefix-list** *<name>* **seq** *<number>* **[deny | permit]** *<network ip/length>* **[ge | le]** *<value>* OR **ipv6 prefix-list** *<name>* **seq** *<number>* **[deny | permit]** *<ipv6 address/prefix-length>* **[ge | le]** *<value>*	Creates a prefix list to define the routes from which a community is to be deleted. Range is **1** to **4294967294**.
Step 2	(config)#	**community-list** *<name>*	Creates a community list for a BGP route map use. The communities defined in this list will be deleted.
Step 3	(config-comm-list)#	**permit [***<value>* **| internet | local-as | no-advertise | no-export]**	Adds an entry to the community list that defines the community string in the community attribute that should be deleted.
Step 4	(config-comm-list)#	**exit**	Exits the Community List Configuration mode and returns to the Global Configuration mode.
Step 5	(config)#	**route-map** *<name>* **[deny | permit]** *<number>*	Creates a route map if one does not already exist. Range is **1** to **4294967295**.
Step 6	(config-route-map)#	**match ip address prefix-list** *<name>* OR **match ipv6 address prefix-list** *<name>*	Specifies the routes from which a community is to be deleted. Reference the *<name>* of the prefix list created in Step 1 to add it to the route map entry.
Step 7	(config-route-map)#	**set comm-list** *<name>* **delete**	Specifies a list of communities to delete. Reference the *<name>* of the community list created in Step 2.
Step 8	(config-route-map)#	**exit**	Exits Route Map Configuration mode and returns to the Global Configuration mode.

Table 11. Deleting a Community *(Continued)*

	Prompt	Command	Description
Step 9	(config)#	**router bgp** *<AS number>*	Enters BGP Configuration mode. Range is **1** to **4294967295**.
Step 10	(config-bgp)#	**address-family [ipv4 | ipv6]**	Enters BGP AF Configuration mode.
Step 11	(config-bgp-ipv4)# (config-bgp-ipv6)#	**neighbor [***<ipv4 address>* | *<ipv6 address>***]**	Enters BGP AF Neighbor Configuration mode for the neighbor to which the route map is to be applied.
Step 12	(config-bgp-ipv4-neighbor)# (config-bgp-ipv6-neighbor)#	**send-community standard**	Enables this AOS device to accept a community attribute or add a community attribute to any advertisement sent by this peer.
Step 13	(config-bgp-ipv4-neighbor)# (config-bgp-ipv6-neighbor)#	**route-map** *<name>* **in**	Applies the route map to the BGP neighbor as an inbound policy.
Step 14	(config-bgp-ipv4-neighbor)# (config-bgp-ipv6-neighbor)#	**do write**	Saves the configuration.

NOTE *Refer to Prefix List on page 35 for more information on creating prefix lists. Refer to Step 4: Deleting Communities from a Route on page 32 for more information on the **set comm-list delete** command.*

Troubleshooting

After configuring BGP, several different commands can be issued from Enable mode in the CLI to assist in troubleshooting. The following tables contain the **show** and **debug** commands that are implemented specifically for BGP.

Table 12. Viewing BGP Information

Command	Description
#**show as-path-list** <*name*>	Displays any AS path lists that have been configured in the AOS device, along with any permit and deny clauses in each list. Optionally, enter the <*name*> of a specific AS path list to display only the list matching the specified AS path list name.
#**show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf** <*name*> **ipv4 | vrf** <*name*> **ipv6]**	Displays the local router ID and AS, plus detailed information about all BGP routes. Information provided for each route includes: origin, destination, next hop, AS path, and whether selected as best. You must specify that the displayed information is for all VRFs and IPv4 route info (**any-vrf ipv4**), all VRFs and IPv6 route info (**any-vrf ipv6**), IPv4 route info (**ipv4**), IPv6 route info (**ipv6**), or for a specific VRF and IPv4 route info (**vrf** <*name*> **ipv4**) or a specific VRF and IPv6 route info (**vrf** <*name*> **ipv6**).
#**show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf** <*name*> **ipv4 | vrf** <*name*> **ipv6] [**<*ip address*> **[**<*subnet mask*>**]] [**<*ipv6 address/prefix-length*>**]**	Displays BGP information about a specific route, including: advertising router IPv4 or IPv6 address, advertising router ID, and neighbors to which the route is advertised. You must specify that the displayed information is for all VRFs and IPv4 route info (**any-vrf ipv4**), all VRFs and IPv6 route info (**any-vrf ipv6**), IPv4 route info (**ipv4**), IPv6 route info (**ipv6**), or for a specific VRF and IPv4 route info (**vrf** <*name*> **ipv4**) or a specific VRF and IPv6 route info (**vrf** <*name*> **ipv6**).
#**show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf** <*name*> **ipv4 | vrf** <*name*> **ipv6] summary**	Displays summarized BGP information. This includes the local router ID and AS, the number of paths received and the number of BGP attribute entries. All BGP neighbors are summarized in a table that displays the remote ID, remote AS, and the number of messages sent and received. You must specify that the displayed information is for all VRFs and IPv4 route info (**any-vrf ipv4**), all VRFs and IPv6 route info (**any-vrf ipv6**), IPv4 route info (**ipv4**), IPv6 route info (**ipv6**), or for a specific VRF and IPv4 route info (**vrf** <*name*> **ipv4**) or a specific VRF and IPv6 route info (**vrf** <*name*> **ipv6**).
#**show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf** <*name*> **ipv4 | vrf** <*name*> **ipv6] community [**<*number*> **| exact | internet | local-as | no-advertise | no-export]**	Displays routes known by the router that contain the specified <*number*>, same community (**exact**), or well-known community string: **internet, local-as, no-advertise,** or **no-export** in their community attribute. You must specify that the displayed information is for all VRFs and IPv4 route info (**any-vrf ipv4**), all VRFs and IPv6 route info (**any-vrf ipv6**), IPv4 route info (**ipv4**), IPv6 route info (**ipv6**), or for a specific VRF and IPv4 route info (**vrf** <*name*> **ipv4**) or a specific VRF and IPv6 route info (**vrf** <*name*> **ipv6**).

Table 12. Viewing BGP Information *(Continued)*

Command	Description
#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf *<name>* **ipv4 | vrf** *<name>* **ipv6] community-list** *<name>* **[exact]**	Displays the BGP routes that are permitted by the specified community list. The optional **exact** keyword restricts the routes displayed to only those whose community lists exactly match those specified in the named community list. If this parameter is omitted, all routes matching any part of the specified community list will be displayed. You must specify that the displayed information is for all VRFs and IPv4 route info (**any-vrf ipv4**), all VRFs and IPv6 route info (**any-vrf ipv6**), IPv4 route info (**ipv4**), IPv6 route info (**ipv6**), or for a specific VRF and IPv4 route info (**vrf** *<name>* **ipv4**) or a specific VRF and IPv6 route info (**vrf** *<name>* **ipv6**).
#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf *<name>* **ipv4 | vrf** *<name>* **ipv6] neighbors [***<ipv4 address>* **|** *<ipv6 address>***]**	Displays information for all BGP neighbors. Information for each neighbor includes: neighbor IPv4 or IPv6 address, neighbor ID, remote AS, settings for BGP intervals, connection status, number of messages, and the local BGP interface IPv4 or IPv6 address. Optionally, limit the information displayed by entering the *<ip address>* of a specific neighbor. You must specify that the displayed information is for all VRFs and IPv4 route info (**any-vrf ipv4**), all VRFs and IPv6 route info (**any-vrf ipv6**), IPv4 route info (**ipv4**), IPv6 route info (**ipv6**), or for a specific VRF and IPv4 route info (**vrf** *<name>* **ipv4**) or a specific VRF and IPv6 route info (**vrf** *<name>* **ipv6**).
#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf *<name>* **ipv4 | vrf** *<name>* **ipv6] neighbors [***<ipv4 address>* **|** *<ipv6 address>***] advertised-routes**	Displays all BGP routes being advertised to the specified neighbor. You must specify that the displayed information is for all VRFs and IPv4 route info (**any-vrf ipv4**), all VRFs and IPv6 route info (**any-vrf ipv6**), IPv4 route info (**ipv4**), IPv6 route info (**ipv6**), or for a specific VRF and IPv4 route info (**vrf** *<name>* **ipv4**) or a specific VRF and IPv6 route info (**vrf** *<name>* **ipv6**).
#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf *<name>* **ipv4 | vrf** *<name>* **ipv6] neighbors [***<ipv4 address>* **|** *<ipv6 address>***] received-routes**	Displays all routes (accepted and rejected) advertised by the specified neighbor. You must specify that the displayed information is for all VRFs and IPv4 route info (**any-vrf ipv4**), all VRFs and IPv6 route info (**any-vrf ipv6**), IPv4 route info (**ipv4**), IPv6 route info (**ipv6**), or for a specific VRF and IPv4 route info (**vrf** *<name>* **ipv4**) or a specific VRF and IPv6 route info (**vrf** *<name>* **ipv6**).
#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf *<name>* **ipv4 | vrf** *<name>* **ipv6] neighbors [***<ipv4 address>* **|** *<ipv6 address>***] routes**	Displays all accepted received routes advertised by the specified neighbor. Routes displayed have passed inbound filtering. You must specify that the displayed information is for all VRFs and IPv4 route info (**any-vrf ipv4**), all VRFs and IPv6 route info (**any-vrf ipv6**), IPv4 route info (**ipv4**), IPv6 route info (**ipv6**), or for a specific VRF and IPv4 route info (**vrf** *<name>* **ipv4**) or a specific VRF and IPv6 route info (**vrf** *<name>* **ipv6**).

Table 12. Viewing BGP Information *(Continued)*

Command	Description
#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf *<name>* **ipv4 | vrf** *<name>* **ipv6] regexp** *<expression>*	Displays a summary of the BGP route table that includes routes whose AS path matches the specified expression. You must specify that the displayed information is for all VRFs and IPv4 route info (**any-vrf ipv4**), all VRFs and IPv6 route info (**any-vrf ipv6**), IPv4 route info (**ipv4**), IPv6 route info (**ipv6**), or for a specific VRF and IPv4 route info (**vrf** <*name*> **ipv4**) or a specific VRF and IPv6 route info (**vrf** <*name*> **ipv6**).
#show running-config router bgp [verbose | *<AS number>***]**	Displays only the BGP configuration of the router. You can optionally limit BGP configuration information by specific AS number, or include detailed information (**verbose**).

*The output of all **show** commands can be limited by appending the following modifiers to the end of the command: | **begin** <text>, | **exclude** <text>, and | **include** <text>. The include modifier limits output to lines that contain the specified text, the **exclude** modifier excludes any lines with the specified text, and the **begin** modifier displays the first line of output with the specified text and all lines thereafter.*

Table 13. Viewing BGP Messages

Command	Description
#debug bgp	Displays general BGP events, such as sent/received message summaries, route processing actions, and results. Keepalive packets are not debugged with this command.
#debug bgp events	Displays significant BGP events, such as neighbor state changes.
#debug bgp [in | out]	Displays the same information as **debug bgp**, but limits messages to the specified direction.
#debug bgp keepalives	Displays BGP keepalive packets.
#debug bgp updates [quiet]	Displays detailed information on BGP updates for all neighbors. The optional **quiet** keyword displays a one-line summary with information about BGP neighbor updates.
#debug bgp scan [database | route-table | soft-reset]	Displays BGP background scan details. You can limit output by displaying only database scan details (**database**), route table scan details (**route-table**), or soft reset scan details (**soft-reset**).

*The **bgp log-neighbor-changes** command can be issued to log state changes of a BGP neighbor. Refer to Log Neighbor Changes on page 19 for more information.*

Table 14. Clearing BGP Statistics

Command	Description
clear bgp *	Clears all BGP neighbors.
clear bgp any-vrf [ipv4 | ipv6] [* | *<number>* **|** *<ipv4 address>* **|** *<ipv6 address>***] [in | out | soft]**	Clears connections for all VRFs.
clear bgp ipv4 [* | *<number>* **|** *<ipv4 address>***] [in | out | soft]**	Clears all BGP IPv4 route information.
clear bgp ipv6 [* | *<number>* **|** *<ipv6 address>***] [in | out | soft]**	Clears all BGP IPv6 route information.
clear vrf *<name>* **[ipv4 | ipv6] [* |** *<number>* **|** *<ipv4 address>* **|** *<ipv6 address>***] [in | out | soft]**	Clears connections for a nondefault VRF.

Show Commands

Show IP AS-Path List

Use the **show as-path-list** command to display any AS path lists that have been configured in the AOS device, along with any permit and deny clauses in each list.

#show as-path-list *<name>*

> *<name>* Optional. Specifies that the command display only the list matching the specified AS path list name. If not specified, all AS path lists are displayed.

In the following example, all AS path lists defined in the AOS device are displayed:

```
#show as-path-list
as-path-list ASPATHLIST1:
  permit 100
  permit 200
  permit 300
  deny 6500
as-path-list ASPATHLIST2:
  permit 400
  permit 500
```

In the following example, only the AS path list with the name ASPATHLIST2 is displayed:

```
#show as-path-list ASPATHLIST2
as-path-list ASPATHLIST2:
  permit 400
  permit 500
```

Show BGP

Use the **show bgp** command to display details about the specified route, including the advertising router IP address, router ID, and the list of neighbors to which this router is being advertised.

#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf *<name>* **ipv4 | vrf** *<name>* **ipv6] [***<ipv4 address> <subnet mask>* **|** *<ipv6 address/prefix-length>***]**

any-vrf ipv4 Displays information (including IPv4 route information) for all VRFs.

any-vrf ipv6 Displays information (including IPv6 route information) for all VRFs.

ipv4 Displays IPv4 route information.

ipv6 Displays IPv6 route information.

vrf <*name*> **ipv4** Displays information for a specific nondefault (named) VRF (including IPv4 route information).

vrf <*name*> **ipv6** Displays information for a specific nondefault (named) VRF (including IPv6 route information).

<*ipv4 address*> Optional. Specifies a valid IPv4 address. IPv4 addresses should be expressed in dotted decimal notation (for example, **10.10.10.1**).

<*subnet mask*> Optional. Specifies the subnet mask that corresponds to a range of IPv4 addresses (network) or a specific host. Subnet masks can be expressed in dotted decimal notation (for example, **255.255.255.0**) or as a prefix length (for example, /**24**).

<*ipv6 address/prefix-length*> Optional. Specifies a valid IPv6 address and prefix. IPv6 addresses should be expressed in colon hexadecimal format (for example, **2001:DB8:3F::/48**).

The following example shows detailed output of the **show bgp ipv4** command:

```
#show bgp ipv4
BGP local router ID is 192.100.8.2, local AS is 66000.
Status codes: * valid, > best, i - internal, o - local
Origin codes: i - IGP, e - EGP, ? - incomplete
Network              NextHop          Metric LocPrf     Path
*>i192.100.5.0/24    192.100.7.1      100               i
*>i192.100.6.0/24    192.100.8.1      100               i
*>o192.100.7.0/24    0.0.0.0                            i
* i192.100.7.0/24    192.100.7.1      100               i
*>o192.100.8.0/24    0.0.0.0                            i
* i192.100.8.0/24    192.100.8.1      100               i
*>i192.100.9.0/24    192.100.7.1      100               i
*>i192.100.9.0/24    192.100.8.1      100               i
*>o192.100.10.0/24   0.0.0.0                            i
*  192.100.10.0/24   192.100.10.2                       65534 i
*> 192.100.11.0/24   192.100.10.2                       65534 i
Total RIB entries = 11
```

The exact prefixes that are being transmitted and received are shown in this output. The o in front of the 192.100.6.0/24 route indicates that the route was injected by the AOS device into BGP from an external route source, such as static, connected, RIP, or OSPF routes.

Use the **show bgp summary** command to display a summary of the BGP route table.

#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf <*name*> ipv4 | vrf <*name*> ipv6] summary

any-vrf ipv4 Displays information (including IPv4 route information) for all VRFs.

any-vrf ipv6 Displays information (including IPv6 route information) for all VRFs.

ipv4 Displays IPv4 route information.

ipv6 Displays IPv6 route information.

vrf <*name*> **ipv4** Displays information for a specific nondefault (named) VRF (including IPv4 route information).

vrf <*name*> **ipv6** Displays information for a specific nondefault (named) VRF (including IPv6 route information).

The following sample output of the **show bgp ipv4 summary** command shows a summarized list of the configured BGP neighbors, as well as their status and statistics.

```
#show ip bgp ipv4 summary
BGP router identifier 192.100.8.2, local AS number 6600
3 network entries, 3 paths, and 3 BGP path attribute entries
    Neighbor      V   AS     MsgRcvd   MsgSent   InQ  OutQ  Up/Down     State/PfxRcd
    192.100.7.1   4   66000  8712        8445      0    0   5d20h39m        3
    192.100.8.1   4   66000  1057        1036      0    0   16:38:49        3
    192.100.10.2  4   65534  8443        8760      0    0   5d20h39m        2
```

Show BGP Community

Use the **show bgp community** command to display only those routes learned via BGP that match the community numbers specified in the command. If no communities are specified, all BGP routes containing a community attribute are shown.

#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf <*name*> ipv4 | vrf <*name*> ipv6] community [<*number*> | exact | internet | local-as | no-advertise | no-export]

any-vrf ipv4 Displays information (including IPv4 route information) for all VRFs.

any-vrf ipv6 Displays information (including IPv6 route information) for all VRFs.

ipv4 Displays IPv4 route information.

ipv6 Displays IPv6 route information.

vrf <*name*> **ipv4** Displays information for a specific nondefault (named) VRF (including IPv4 route information).

vrf <*name*> **ipv6** Displays information for a specific nondefault (named) VRF (including IPv6 route information).

<*number*> Optional. Displays routes that contain this value in their community attribute. This is a numeric value that can be an integer from **1** to **4294967295** or string in the form aa:nn, where the value of aa is the AS number and the value of *nn* is the desired local preference to be used in the service provider network. Multiple community-number parameters can be present in the command.

exact Optional. Displays routes that contain the same community.

internet Optional. Displays routes that contain this value in their community attribute. This represents the well-known reserved community string INTERNET.

local-as Optional. Displays routes that contain this value in their community attribute. This represents the well-known reserved community string NO_EXPORT_SUBCONFED. Routes containing this attribute should not be advertised to external BGP peers.

no-advertise Optional. Displays routes containing this value in the community attribute. This represents the well-known reserved community string NO_ADVERTISE. Routes containing this attribute should not be advertised to any BGP peer.

no-export Optional. Displays routes containing this value in the community attribute. This represents the well-known reserved community number for NO_EXPORT. Routes containing this attribute should not be advertised to BGP peers outside a confederation boundary.

In the following example, all BGP routes are displayed whose community attributes match those listed in the **show bgp community** command.

```
#show bgp ipv4 community 10:999
BGP local router ID is 192.100.12.1, local AS is 1.
Status codes: * valid, > best, i - internal, o - local
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network           Next Hop       Metric     LocPrf     Path
*  192.100.5.0/24     192.100.5.2                          66000 i
*  192.100.6.0/24     192.100.5.2                          66000 i
*> 192.100.7.0/24     192.100.5.2                          66000 i
*> 192.100.8.0/24     192.100.5.2                          66000 i
*> 192.100.9.0/24     192.100.5.2                          66000 i
*> 192.100.10.0/24    192.100.12.2                         65534 i
*  192.100.10.0/24    192.100.5.2                          66000 i
*  192.100.10.0/24    192.100.6.2                          66000 i
Total RIB entries = 8
```

Information displayed includes: the ID of this router and its AS number; the destination Network address of the route learned; the Next-Hop address to that network; the Metric; the Local Preference (LocPrf) value; and the AS Path to the destination network.

Show BGP Community-List

Use the **show bgp community-list** command to display BGP routes that are permitted by the specified community list.

#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf <*name*> **ipv4 | vrf** <*name*> **ipv6] community-list** <*name*> **[exact]**

any-vrf ipv4 Displays information (including IPv4 information) for all VRFs.

any-vrf ipv6 Displays information (including IPv6 information) for all VRFs.

ipv4 Displays IPv4 route information.

ipv6 Displays IPv6 route information.

vrf <*name*> **ipv4** Displays information for a specific nondefault (named) VRF (including IPv4 route information).

vrf <*name*> **ipv6** Displays information for a specific nondefault (named) VRF (including IPv6 route information).

<*name*> Specifies the name of the community list whose routes are to be displayed.

exact Optional. Restricts the routes displayed to only those whose community lists exactly match those specified in the named community list. If this parameter is omitted, all routes matching any part of the specified community list will be displayed.

In the following example, all BGP routes are displayed whose community number match those defined in the community list named CLIST1.

```
#show bgp ipv4 community-list CLIST1
BGP local router ID is 10.22.131.241, local AS is 302.
Status codes: * valid, > best, i - internal, o - local
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        Metric     LocPrf     Path
    10.22.152.20/30    10.22.131.10    304                   302 300 1 3 4 i
    10.22.152.24/29    10.22.131.10    304                   302 300 1 3 4 5 i
    10.22.152.36/30    10.22.131.10    304                   302 300 1 3 4 i
    10.22.152.52/30    10.22.131.10    304                   302 300 1 3 4 i
    11.0.0.0/30        10.22.131.10    304                   302 300 1 3 4 6 i
    12.0.0.0/30        10.22.131.10    304                   302 300 1 3 4 6 i
    13.0.0.0/30        10.22.131.10    304                   302 300 1 3 4 6 i
    14.0.0.0/30        10.22.131.10    304                   302 300 1 3 4 6 i
    20.0.0.0/30        10.22.131.10    304                   302 300 1 3 4 5 i
    21.0.0.0/30        10.22.131.10    304                   302 300 1 3 4 5 i
Total RIB entries = 10
```

Show BGP Neighbors

Use the **show bgp neighbors** command to display information for the specified BGP neighbor.

#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf <*name*> ipv4 | vrf <*name*> ipv6] neighbors [<*ipv4 address*> | <*ipv6 address*>] [advertised-routes | received-routes | routes]

any-vrf ipv4 Displays information (including IPv4 route information) for all VRFs.

any-vrf ipv6 Displays information (including IPv6 route information) for all VRFs.

ipv4 Displays IPv4 route information.

ipv6 Displays IPv6 route information.

vrf <*name*> **ipv4** Displays information for a specific nondefault (named) VRF (including IPv4 route information).

vrf <*name*> **ipv6** Displays information for a specific nondefault (named) VRF (including IPv6 route information).

<ipv4 address> Optional. Displays information for the specified neighbor. IPv4 addresses should be expressed in dotted decimal notation (for example, **10.10.10.1**). If no IPv4 address is entered, information for all neighbors is displayed.

<ipv6 address> Optional. Displays information for the specified neighbor. IPv6 addresses are expressed in colon hexadecimal format (for example, **2001:DB8:1::1**). If no IPv6 address is entered, information for all neighbors is displayed.

advertised-routes Optional. Displays all routes being advertised to the specified neighbor. Command output is the same as for **show ip bgp** except filtered to only the BGP routes being advertised to the specified neighbor.

received-routes Optional. Displays all routes (accepted and rejected) advertised by the specified neighbor. Routes can be rejected by inbound filters, such as prefix list filters.

routes Optional. Displays all accepted received routes advertised by the specified neighbor. Routes displayed have passed inbound filtering. This command output is the same as **show ip bgp** except the output is filtered to those learned from the specified neighbor.

> NOTE *Entries that are not filtered by a local BGP policy are marked with an asterisk (*) to show they are valid. Entries that are deemed the best path to an advertised route are marked with a caret (>).*

The following examples show a couple of output variations of the **show bgp ipv4 neighbors** command:

```
#show bgp ipv4 neighbors
BGP neighbor is 10.15.43.17, remote AS 100, external link
Configured hold time is 180, keepalive interval is 60 seconds
Default minimum time between advertisement runs is 30 seconds
Connections established 6; dropped 5
Last reset: Interface went down
  Connection ID: 15
    BGP version 4, remote router ID 8.1.1.1
    BGP state is Established, for 01:55:05
    Negotiated hold time is 180, keepalive interval is 60 seconds
    Message statistics:
      InQ depth is 0, OutQ depth is 0
                        Sent            Rcvd
      Opens:            1               1
      Notifications:    0               0
      Updates:          0               8
      Keepalives:       116             116
      Unknown:          0               0
      Total:            117             125
Local host: 10.15.43.18, Local port: 179
Foreign host: 10.15.43.17, foreign port: 1048
  Flags: passive open
```

```
#show bgp ipv4 neighbors 10.15.43.34 advertised-routes
BGP local router ID is 10.0.0.1, local AS is 101.
Status codes: * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
     Network      NextHop          Metric Path
*>   1.0.0.0/8    10.15.43.17      1 100 i
*>   2.0.0.0/9    10.15.43.17      1 100 i
```

Show BGP Regexp

Use the **show bgp regexp** command to display a summary of the BGP route table that includes routes whose AS path matches the specified expression.

#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf *<name>* **ipv4 | vrf** *<name>* **ipv6] regexp** *<expression>*

any-vrf ipv4 Displays information (including IPv4 route information) for all VRFs.

any-vrf ipv6 Displays information (including IPv6 route information) for all VRFs.

ipv4 Displays IPv4 route information.

ipv6 Displays IPv6 route information.

vrf *<name>* **ipv4** Displays information for a specific nondefault (named) VRF (including IPv4 route information).

vrf *<name>* **ipv6** Displays information for a specific nondefault (named) VRF (including IPv6 route information).

<expression> Specifies the regular expression to match when displaying BGP routes. Only those routes whose AS path matches this expression will be displayed in the output.

> NOTE *Regular expressions are strings of characters used in BGP to identify routes by their AS path. Refer to AS Regular Expressions on page 110 for a detailed list of valid AS regular expressions.*

> NOTE *Entries that are not filtered by a local BGP policy are marked with an asterisk (*) to show they are valid. Entries that are deemed the best path to an advertised route are marked with a caret (>).*

The following sample output of the **show bgp ipv4 regexp\b303\b** command shows all of the entries in the BGP database that contain **303** in the AS path.

```
#show bgp ipv4 regexp \b303\b
BGP local router ID is 192.168.3.1, local AS is 304.
Status codes: * valid, > best, i - internal, o - local
Origin codes: i - IGP, e - EGP, ? - incomplete
Network              NextHop Metric       LocPrf Path
10.22.130.8/29       10.22.132.9          303 304 302 i
* i10.22.130.240/28  10.22.132.1          100 303 300 i
* 10.22.130.240/28   10.22.132.9          303 300 i
10.22.131.0/29       10.22.132.9          303 304 302 i
10.22.131.8/29       10.22.132.9          303 304 302 i
```

```
* i10.22.131.16/29    10.22.132.1     0 100 303 i
* 10.22.131.16/29     10.22.132.9     0 303 i
* i10.22.131.240/28   10.22.132.1     100 303 300 i
* 10.22.131.240/28    10.22.132.9     303 300 i
* 10.22.132.0/29      10.22.131.1     0 302 303 i
* 10.22.132.0/29      10.22.131.9     0 302 303 i
* i10.22.132.0/29     10.22.132.1     0 100 303 i
*> 10.22.132.0/29     10.22.132.9     0 303 i
* 10.22.132.8/29      10.22.131.1     0 302 303 i
* 10.22.132.8/29      10.22.131.9     0 302 303 i
* 10.22.132.8/29      10.22.132.9     0 303 i
* i10.22.132.240/28   10.22.132.1     0 100 303 i
*> 10.22.132.240/28   10.22.132.9     0 303 i
10.22.134.0/29        10.22.132.9     303 304 i
10.22.134.8/29        10.22.132.9     303 304 i
10.22.134.16/29       10.22.132.9     303 304 i
10.22.134.24/29       10.22.132.9     303 304 i
10.22.134.32/29       10.22.132.9     303 304 i
10.22.134.40/29       10.22.132.9     303 304 i
10.22.134.48/29       10.22.132.9     303 304 i
10.22.134.56/29       10.22.132.9     303 304 i
10.22.134.64/29       10.22.132.9     303 304 i
10.22.134.80/29       10.22.132.9     303 304 i
10.22.135.0/29        10.22.132.9     303 304 305 i
10.22.135.8/29        10.22.132.9     303 304 305 i
Total RIB entries = 30
```

Show Running-Config Router BGP

Use the **show running-config router bgp** command to display only the BGP portion of the running configuration of the router.

#show running-config router bgp

Whereas the **show running-config** command displays all the nondefault parameters contained in the current running configuration file, the **show running-config router bgp** is helpful when a user would like to view only the BGP portion of the running configuration. This command is particularly useful when the running configuration is long and scrolls through many screens.

> *Route maps, prefix-lists, and other lists or filters that are being used with BGP are not displayed in the output of the* ***show running-config router bgp*** *command.*

The following sample output of the **show running-config router bgp** command shows the BGP portion of a running configuration:

```
#show run router bgp
Building configuration...
!
!
router bgp 6500
    neighbor 192.168.100.1
        remote-as 6510
        no shutdown
        exit
    address-family ipv4
        network 192.168.5.0 mask 255.255.255.0
        neighbor 192.168.100.1
        no default-originate
        soft-reconfiguration inbound
        route-map FILTER out
        no shutdown
        exit
    vrf red
        neighbor 10.200.1.20
            remote-as 6520
            no shutdown
            exit
        address-family ipv4
            neighbor 10.200.1.20
            soft-reconfiguration inbound
            no shutdown
            exit
        exit
    exit
!
end
```

Debug Commands

Debug BGP

Use the **debug bgp** command to activate debug messages associated with BGP. Debug messages are displayed in real time. Use the no form of this command to disable the debug messages.

> NOTE *Turning on a large amount of debug information can adversely affect the performance of your unit.*

#debug bgp [events | in | out | keepalives | updates | updates quiet | scan | scan soft-reset | scan database | scan route-table]

events Optional. Displays significant BGP events, such as a neighbor state change.

in/out Optional. Displays the same information as the **debug ip bgp** command, but limits messages to the specified direction (in or out).

keepalives Optional. Displays BGP keepalive packets.

updates Optional. Displays detailed information on BGP updates for all neighbors.

updates quiet Optional. Displays summary information about BGP neighbor updates. (Note: **updates quiet** displays a one-line summary of the output that is displayed when **debug ip bgp updates** is issued.)

> *If no arguments are given, the* ***debug ip bgp*** *command displays general BGP events, such as sent/received message summaries, route processing actions, and results. Keepalive packets are not debugged with this command.*

scan | scan soft-reset | scan database | scan route-table Optional. Displays BGP background scan details. These details can be limited to database, route table, or soft-reset scan details.

The following example enables debug messages on general outbound BGP messages and events:

```
#debug bgp out
2011.09.08 09:50:53 BGP.LOG VRF: red neighbor 192.101.10.2 Down - User reset

2011.09.08 09:50:53 BGP.OUT VRF: red 192.101.7.1[6]: Transmitting msg, type=UPDATE (2), len=27
2011.09.08 09:50:53 BGP.OUT VRF: red 192.101.7.1[6]: Transmitting msg, type=UPDATE (2), len=675
2011.09.08 09:50:53 BGP.OUT VRF: red 192.101.8.1[7]: Transmitting msg, type=UPDATE (2), len=27
2011.09.08 09:50:53 BGP.OUT VRF: red 192.101.8.1[7]: Transmitting msg, type=UPDATE (2), len=675
```

The following is sample output from the **debug bgp scan** command:

```
#debug bgp scan
2011.09.08 10:52:52 BGP.SCAN RT VRF: red Scanning route table entry: 127.0.0.0/8 (0x44c82fb0) FWD
 valid: 1, dist: 0, metric: [0, 0]
 sprio: 10, add-seq: 2, chg-seq: 3

2011.09.08 10:52:52 BGP.SCAN RT VRF: red Route table entry (0x44c82fb0) remove
2011.09.08 10:52:52 BGP.SCAN RT VRF: red Scanning route table entry: 192.101.1.0/24 (0x44c544e0)
    FWD
 valid: 1, dist: 200, metric: [0, 0]
 sprio: 7, add-seq: 1043, chg-seq: 1734

2011.09.08 10:52:52 BGP.SCAN RT VRF: red Route table entry (0x44c544e0) advertise
2011.09.08 10:52:52 BGP.SCAN RT VRF: red Found existing BGP entry (0x44c544e0)
2011.09.08 10:52:52 BGP.SCAN RT VRF: red Skipping BGP entry: 192.101.1.0/24 (45fb12f4)
 valid: 1, best: 1, local: 0, route table: 1, tag: 9, route: 44c544e0
 path attrib info (45ac3a90): 1
   rib: 1, removal: 0, path len: 1, origin: 0
   nexthop: 192.101.5.1, local pref: 100 (set), med: 4294967295, aa: 0
   communities: none
 received from :
  peer: 192.101.7.1, router-id: 192.101.9.1
  advertised to:
    peer: 192.101.10.2, router-id: 192.101.11.254
  withdrawn from:
2011.09.08 10:52:52 BGP.SCAN RT VRF: red Scanning route table entry: 192.101.2.0/24 (0x44c7d5c8)
      FWD
  valid: 1, dist: 200, metric: [0, 0]
  sprio: 7, add-seq: 183, chg-seq: 178

2011.09.08 10:52:52 BGP.SCAN RT VRF: red Route table entry (0x44c7d5c8) advertise
2011.09.08 10:52:52 BGP.SCAN RT VRF: red Found existing BGP entry (0x44c7d5c8)
2011.09.08 10:52:52 BGP.SCAN RT VRF: red Skipping BGP entry: 192.101.2.0/24 (45fc7638)
  valid: 1, best: 1, local: 0, route table: 1, tag: 9, route: 44c7d5c8
  path attrib info (45ac3a90): 1
     rib: 1, removal: 0, path len: 1, origin: 0
     nexthop: 192.101.5.1, local pref: 100 (set), med: 4294967295, aa: 0
     communities: none
  received from :
    peer: 192.101.7.1, router-id: 192.101.9.1
  advertised to:
    peer: 192.101.10.2, router-id: 192.101.11.254
  withdrawn from:
2011.09.08 10:52:52 BGP.SCAN RT VRF: red Scanning route table entry: 192.101.3.0/24 (0x44c7d520)
      FWD
  valid: 1, dist: 200, metric: [0, 0]
  sprio: 7, add-seq: 184, chg-seq: 180
```

Clear Commands

Clear BGP

BGP sessions must be cleared for BGP policy changes, such as alterations to prefix list filters or actions taken by a route map, to take effect. Use the **clear bgp** command to clear BGP neighbors as specified:

#clear bgp *

* Clears all BGP neighbors.

Clear BGP Any VRF

Use the **clear bgp any-vrf** command to clear BGP connections for all VRFs as specified:

#clear bgp any-vrf [ipv4 | ipv6] [* | *<number>* **|** *<ipv4 address>* **|** *<ipv6 address>***] [in | out | soft]**

ipv4 Clears BGP IPv4 route information.

ipv6 Clears BGP IPv6 route information.

* Clears all BGP neighbors.

<number> Clears all BGP neighbors with the specified AS number. Range is **1** to **4294967295**.

<ipv4 address> Clears the BGP neighbor with the specified IPv4 address. IPv4 addresses should be expressed in dotted decimal notation (for example, **10.10.10.1**).

<ipv6 address> Clears the BGP neighbor with the specified IPv6 address. IPv6 addresses should be expressed in colon hexadecimal format (for example, **2001:DB8:1::1**).

in Causes a soft reset inbound with a neighbor, reprocessing routes advertised by that neighbor.

out Causes a soft reset outbound with a neighbor, resending advertised routes to that neighbor.

soft Causes a soft reset both inbound and outbound.

Clear BGP IPv4

Use the **clear bgp ipv4** command to clear BGP IPv4 route information only on the default VRF as specified:

#clear bgp ipv4 [* | *<number>* | *<ipv4 address>*] [in | out | soft]

* Clears all BGP neighbors.

<number> Clears all BGP neighbors with the specified AS number. Range is **1** to **4294967295**.

<ipv4 address> Clears the BGP neighbor with the specified IPv4 address. IPv4 addresses should be expressed in dotted decimal notation (for example, **10.10.10.1**).

in Causes a soft reset inbound with a neighbor, reprocessing routes advertised by that neighbor.

out Causes a soft reset outbound with a neighbor, resending advertised routes to that neighbor.

soft Causes a soft reset both inbound and outbound.

Clear BGP IPv6

Use the **clear bgp ipv6** command to clear BGP IPv6 route information only on the default VRF as specified:

#clear bgp ipv6 [* | *<number>* | *<ipv6 address>*] [in | out | soft]

* Clears all BGP neighbors.

<number> Clears all BGP neighbors with the specified AS number. Range is **1** to **4294967295**.

<ipv6 address> Clears the BGP neighbor with the specified IPv6 address. IPv6 addresses should be expressed in colon hexadecimal format (for example, **2001:DB8:1::1**).

in Causes a soft reset inbound with a neighbor, reprocessing routes advertised by that neighbor.

out Causes a soft reset outbound with a neighbor, resending advertised routes to that neighbor.

soft Causes a soft reset both inbound and outbound.

Clear BGP VRF

Use the **clear bgp vrf** *<name>* command to clear BGP connections for a nondefault (named) VRF as specified:

#clear bgp vrf *<name>* [ipv4 | ipv6] [* | *<number>* | *<ipv4 address>* | *<ipv6 address>*] [in | out | soft]

ipv4 Specifies that BGP IPv4 connections are cleared on the nondefault (named) VRF.

ipv6 Specifies that BGP IPv6 connections are cleared on the nondefault (named) VRF.

* Clears all BGP neighbors.

<*number*> Clears all BGP neighbors with the specified AS number. Range is **1** to **4294967295**.

<*ipv4 address*> Clears the BGP neighbor with the specified IPv4 address. IPv4 addresses should be expressed in dotted decimal notation (for example, **10.10.10.1**).

<*ipv6 address*> Clears the BGP neighbor with the specified IPv6 address. IPv6 addresses should be expressed in colon hexadecimal format (for example, **2001:DB8:1::1**).

in Causes a soft reset inbound with a neighbor, reprocessing routes advertised by that neighbor.

out Causes a soft reset outbound with a neighbor, resending advertised routes to that neighbor.

soft Causes a soft reset both inbound and outbound.

> NOTE *Typically, soft resets should be used because hard resets can disrupt the network. A hard reset clears the TCP connection with the specified peers, which results in clearing the table. This method of clearing is disruptive and causes peer routers to record a route flap for each route. Refer to Soft Reconfiguration Inbound on page 46 for additional information.*

Strategies for Troubleshooting Specific BGP Problems

An AOS device running BGP might not send or receive the routes that it should for several reasons:

- It cannot communicate with a neighbor
- It is not authorized to transmit, or to accept, the routes in question

View BGP neighbors to make certain the neighbor exists by entering the command **show bgp neighbor** at the Enable mode prompt.

If the AOS device is able to communicate with the neighbor, but it is not receiving the routes that it should, then BGP filters need to be examined.

Removing Filters

BGP allows a user to configure policies to filter routes accepted from and advertised to neighbors. These policies are configured in prefix lists, route maps, and ACLs. Since the policies can be quite complicated, they open room for errors.

One of the first steps in troubleshooting BGP is to remove any inbound or outbound filters from the neighbor. If the AOS device begins receiving or advertising the expected routes, then the conclusion is that the filters are causing the problem. The following commands can be issued from BGP AF Neighbor Configuration mode to remove the appropriate filter:

Remove a prefix list:

```
(config-bgp-ipv4-neighbor)#no prefix-list <name> [in | out]
OR
(config-bgp-ipv6-neighbor)#no prefix-list <name> [in | out]
```

<*name*> Specifies the name of the prefix list to be removed.

in Specifies to remove the inbound filter.

out Specifies to remove the outbound filter.

Remove a route map:

```
(config-bgp-ipv4-neighbor)#no route-map <name> [in | out]
OR
(config-bgp-ipv6-neighbor)#no route-map <name> [in | out]
```

<name> Specifies the name of the route map to be removed.

in Specifies to remove the inbound filter.

out Specifies to remove the outbound filter.

Remove an ACL:

```
(config-bgp-ipv4-neighbor)#no distribute-list <ipv4 acl name> [in | out]
OR
(config-bgp-ipv6-neighbor)#no distribute-list <ipv6 acl name> [in | out
```

<ipv4 acl name> Specifies the name of the IPv4 ACL to be removed.

<ipv6 acl name> Specifies the name of the IPv6 ACL to be removed.

in Specifies to remove the inbound filter.

out Specifies to remove the outbound filter.

Clear the BGP neighbor with a soft reset and observe if the AOS device begins to receive routes. If routes are received, then it is confirmed that the filter is the problem. Reconfigure the prefix list, route map, or ACL keeping in mind that the AOS device processes entries in order by sequence number and stops as soon as it finds a match.

A prefix list or route map can also be monitored to see how it is affecting traffic by viewing the list or map and checking the number of packets the router has matched to it. Refer to *Troubleshooting a Prefix List on page 107* and *Troubleshooting a Route Map on page 108*.

If a prefix list is being used with a route map, then it is important to determine whether it is the prefix list or the route map configuration that has the error. An entry in the prefix list that permits all routes can be configured:

```
(config)#ip prefix-list <name> seq <number> permit 0.0.0.0/0 le 32
OR
(config)#ipv6 prefix-list <name> seq <number> permit 0.0.0.0/0 le 32
```

<name> Specifies the name of the prefix list.

<number> Specifies the entry's unique sequence number that determines the processing order. Lower numbered entries are processed first. Range is **1** to **4294967294**.

Clear the BGP neighbor with a soft reset and observe if the AOS device begins to receive routes. If routes are received, then it can be concluded that the prefix list must be reconfigured. Otherwise, the route map is the problem.

NOTE *When adding an entry to a prefix list to permit all routes, the sequence number specified should be the lower than all other entries contained within the prefix list.*

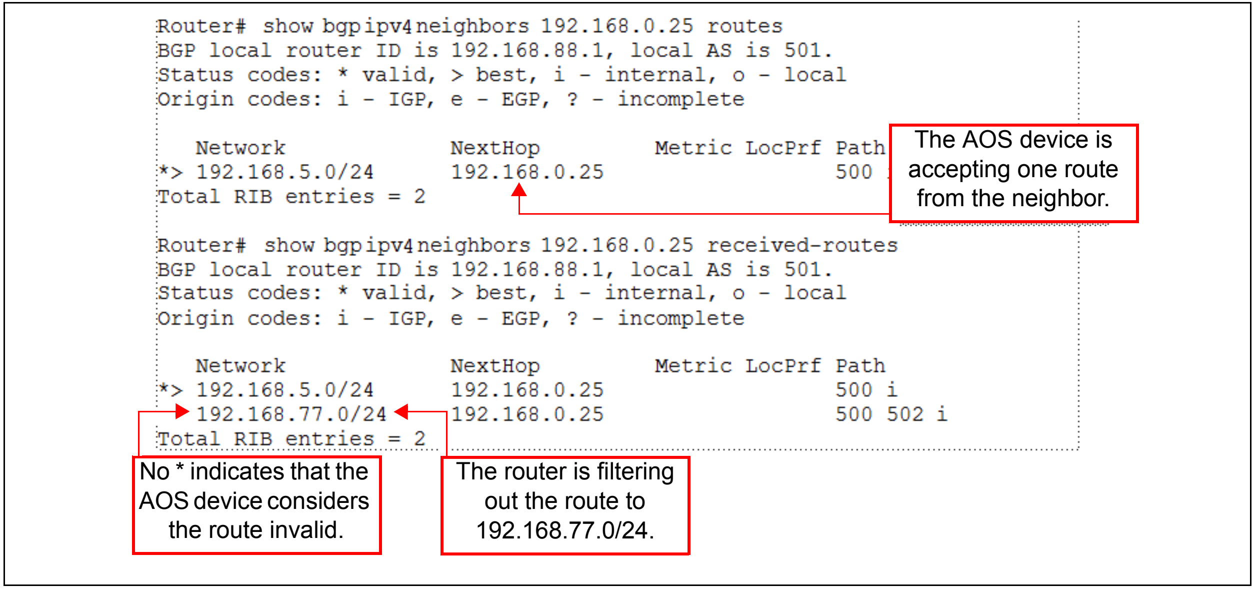
BGP Will Not Accept Routes

If it is suspected that filters are keeping the AOS device from receiving routes, compare the routes that BGP receives from a neighbor to those it actually accepts. Enter:

#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf *<name>* **ipv4 | vrf** *<name>* **ipv6] neighbor [***<ipv4 address>* **|** *<ipv6 address>***] received-routes**

#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf *<name>* **ipv4 | vrf** *<name>* **ipv6] neighbor [***<ipv4 address>* **|** *<ipv6 address>***] routes**

Note any routes that are displayed when the first command is entered, but not displayed when the second command is entered. These routes are being filtered out. Also, determine if the filter is rejecting a route by locating the asterisk (*) in front of the network address. The absence of an asterisk means that the AOS device considers the displayed route invalid. See *Figure 12 on page 104*.



```
Router# show bgpipv4neighbors 192.168.0.25 routes
BGP local router ID is 192.168.88.1, local AS is 501.
Status codes: * valid, > best, i - internal, o - local
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network             NextHop            Metric LocPrf Path
*> 192.168.5.0/24      192.168.0.25                     500
Total RIB entries = 2

Router# show bgpipv4neighbors 192.168.0.25 received-routes
BGP local router ID is 192.168.88.1, local AS is 501.
Status codes: * valid, > best, i - internal, o - local
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network             NextHop            Metric LocPrf Path
*> 192.168.5.0/24      192.168.0.25                     500 i
   192.168.77.0/24     192.168.0.25                     500 502 i
Total RIB entries = 2
```

The AOS device is accepting one route from the neighbor.

No * indicates that the AOS device considers the route invalid.

The router is filtering out the route to 192.168.77.0/24.

Figure 12. Comparing Accepted Routes to All Routes Received

The next step in troubleshooting an interface that will not accept routes is to remove filters from the neighbor (refer to *Removing Filters on page 102*). If the filter is the problem, then troubleshoot it as described in *Troubleshooting a Prefix List on page 107* and *Troubleshooting a Route Map on page 108*.

BGP Cannot Communicate with a Neighbor

Unlike other routing protocols, BGP does not automatically search for and exchange routes with connected routers. Each BGP neighbor must be manually added and configured on the AOS device.

First, view the BGP neighbor and double-check its IP address:

#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf *<name>* **ipv4 | vrf** *<name>* **ipv6] neighbors**

Ping the neighbor and check connectivity.

If the ping is successful, but the AOS device does not seem to be exchanging BGP messages, the maximum hop count for BGP messages may need to be adjusted using the **ebgp-multihop** command. Typically, external neighbors can be reached from a directly connected interface making them one hop away. If they are not, then the number of hops between the interface and the neighbor must be specified. For example:

#ebgp-mulithop 4

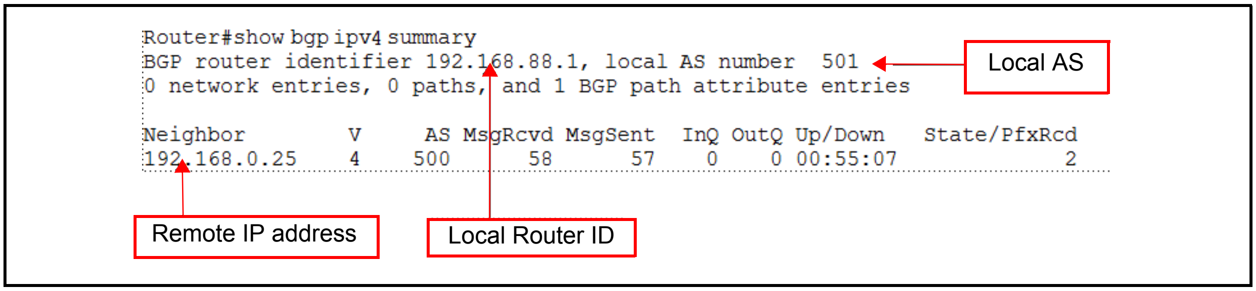
NOTE *A loopback interface adds a hop to the route. Even if the external neighbor is directly connected, **ebgp-multihop** must be enabled if a loopback interface is used as the source BGP interface. Refer to eBGP Multihop on page 20 for additional information on this command.*

Next, record the current settings in the AOS device and verify that they match those that have been agreed upon with the entity that controls the external AS. *Table 15 on page 105* displays the key information that should be verified and how to view the settings on the AOS device.

Table 15. Checking BGP Configuration

Key Information	How to View	Record the AOS Device Setting
local AS	**show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf** *<name>* **ipv4 | vrf** *<name>* **ipv6] [summary]**	
local router ID	**show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf** *<name>* **ipv4 | vrf** *<name>* **ipv6] [summary]**	
local router IP address	**show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf** *<name>* **ipv4 | vrf** *<name>* **ipv6] neighbor**	
neighbor router ID	**show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf** *<name>* **ipv4 | vrf** *<name>* **ipv6] neighbor**	
neighbor IP address	**show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf** *<name>* **ipv4 | vrf** *<name>* **ipv6] neighbor**	
remote AS	**show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf** *<name>* **ipv4 | vrf** *<name>* **ipv6] neighbor**	

Figure 13 below shows sample output and where to locate some of the information for *Table 15 on page 105*.



```
Router#show bgp ipv4 summary
BGP router identifier 192.168.88.1, local AS number  501
0 network entries, 0 paths, and 1 BGP path attribute entries

Neighbor         V      AS MsgRcvd MsgSent  InQ OutQ Up/Down    State/PfxRcd
192.168.0.25     4     500      58      57    0    0 00:55:07              2
```

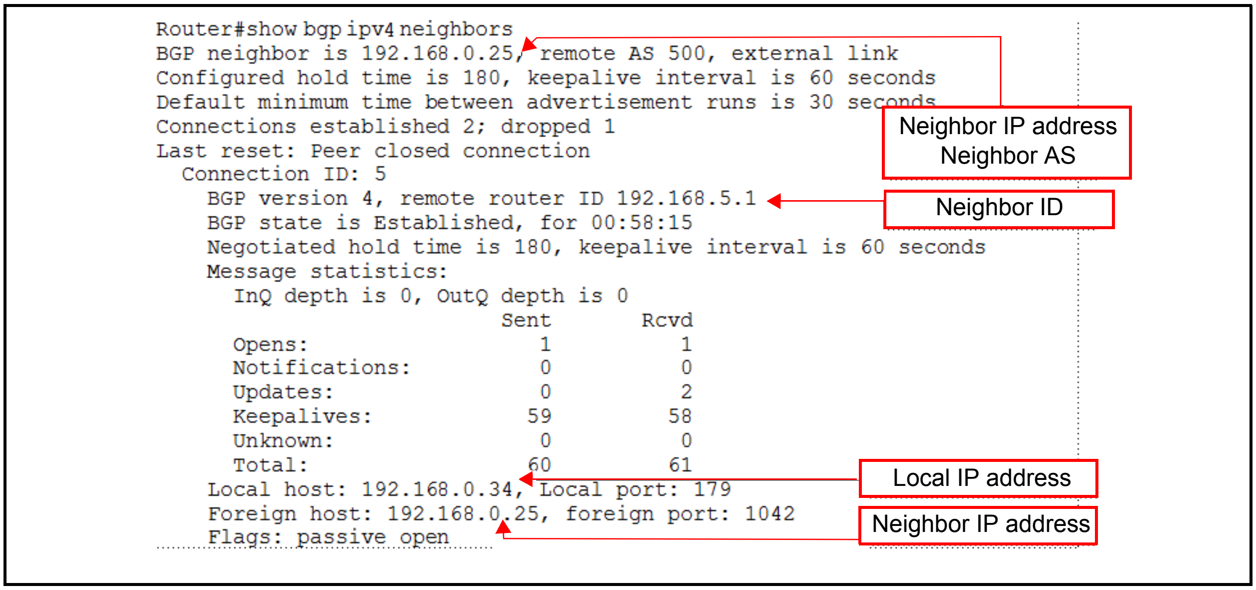
Figure 13. Viewing Local ID and Local AS

When BGP cannot reach the configured neighbor, the following debug messages are received on the console:

```
BGP EVT 1.1.1.1[1]: IDLE->CONNECT
BGP EVT 1.1.1.1[1]: CONNET->IDLE
BGP OUT 1.1.1.1[1]: TCP error 0 connecting to peer (events:connect)
```

In this example, the interface is attempting to connect to a peer through the peer's IPv4 loopback address (1.1.1.1), which the router does not consider to be directly connected.

When configuring the BGP neighbor, it is important to always identify it by the IP address for the connecting interface, even if the remote router uses a different router ID. For example, *Figure 14* displays information about a local router's BGP neighbor. The neighbor uses the IPv4 address 192.168.5.1 for its router ID. However, the remote IPv4 address is 192.168.0.25, and this is the IPv4 address that should be entered when configuring the neighbor.



```
Router#show bgp ipv4 neighbors
BGP neighbor is 192.168.0.25, remote AS 500, external link
Configured hold time is 180, keepalive interval is 60 seconds
Default minimum time between advertisement runs is 30 seconds
Connections established 2; dropped 1
Last reset: Peer closed connection
  Connection ID: 5
    BGP version 4, remote router ID 192.168.5.1
    BGP state is Established, for 00:58:15
    Negotiated hold time is 180, keepalive interval is 60 seconds
    Message statistics:
      InQ depth is 0, OutQ depth is 0
                           Sent       Rcvd
      Opens:                  1          1
      Notifications:          0          0
      Updates:                0          2
      Keepalives:            59         58
      Unknown:                0          0
      Total:                 60         61
    Local host: 192.168.0.34, Local port: 179
    Foreign host: 192.168.0.25, foreign port: 1042
    Flags: passive open
```

Figure 14. Viewing a BGP Neighbor

A BGP Interface Will Not Send Routes to a Neighbor

If a remote host cannot reach the local network, it is possible that the BGP interface may not be sending the remote unit the correct routes. View the routes the AOS device is advertising to the neighbor by entering the following command:

#show bgp [any-vrf ipv4 | ipv4 | vrf <*name*> **ipv4] neighbor** <*ipv4 address*> **advertised-routes**

Verify that BGP has been configured to advertise the network by viewing the running configuration. Also, check outbound filters (both prefix lists and route maps) as you would inbound filters.

If the AOS device still cannot send or receive routes, then it is probably having trouble connecting to the neighbor (refer to *BGP Cannot Communicate with a Neighbor on page 104*).

Troubleshooting a Prefix List

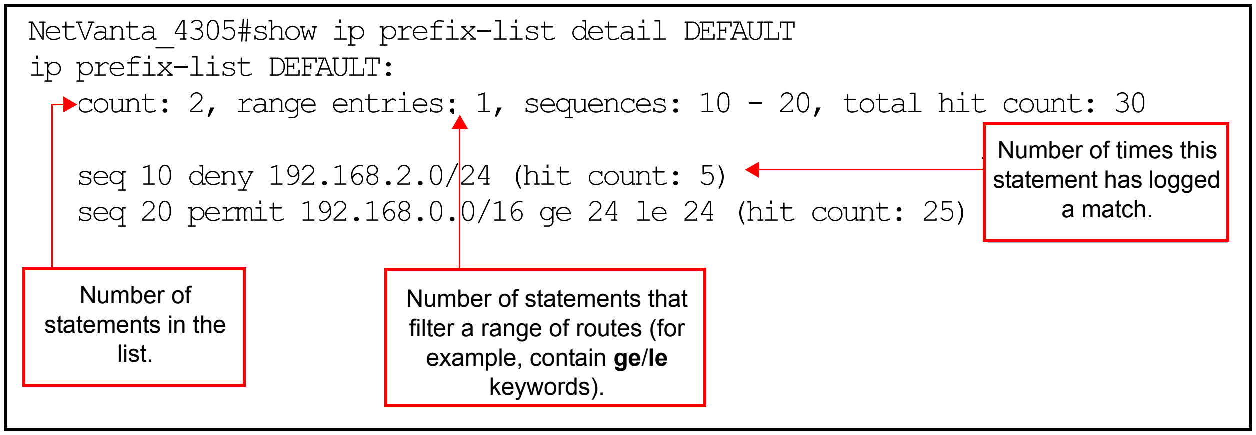
Use the following Enable mode command to view a prefix list:

#show [ip | ipv6] prefix-list [detail | summary] <*name*>

The **ip** and **ipv6** keywords specify whether you are viewing IPv4 or IPv6 prefix lists. The **detail** and **summary** keywords are optional and mutually exclusive. If only the name of the prefix list is entered, then the permit and deny statements are displayed and listed by sequence number. The **summary** keyword produces output that lists the number of statements, their sequence numbers, and the number of these statements that include a range of valid prefixes. The **detail** keyword produces all the information shown by the other two commands, as well as the number of times a BGP prefix has matched each statement.

Check all of the statements for hits. Statements that have been misconfigured often have no hits. If the entire list has no hits, then the list may not have been applied to the BGP neighbor (if the list is applied to a route map, ensure that the map has been applied to the BGP neighbor). Also, verify that the list is correctly applied to either inbound or outbound data.

Figure 15 on page 107 shows sample output of the detailed command.



```
NetVanta_4305#show ip prefix-list detail DEFAULT
ip prefix-list DEFAULT:
   count: 2, range entries: 1, sequences: 10 - 20, total hit count: 30

   seq 10 deny 192.168.2.0/24 (hit count: 5)
   seq 20 permit 192.168.0.0/16 ge 24 le 24 (hit count: 25)
```

Number of times this statement has logged a match.
Number of statements in the list.
Number of statements that filter a range of routes (for example, contain **ge**/**le** keywords).

Figure 15. Viewing a Prefix List

The following are useful tips to keep in mind when searching for misconfigurations in a prefix list:

- If a statement does not include a range of prefixes, then a route must match the statement exactly in order to be selected. Make sure that the prefix length is correct.

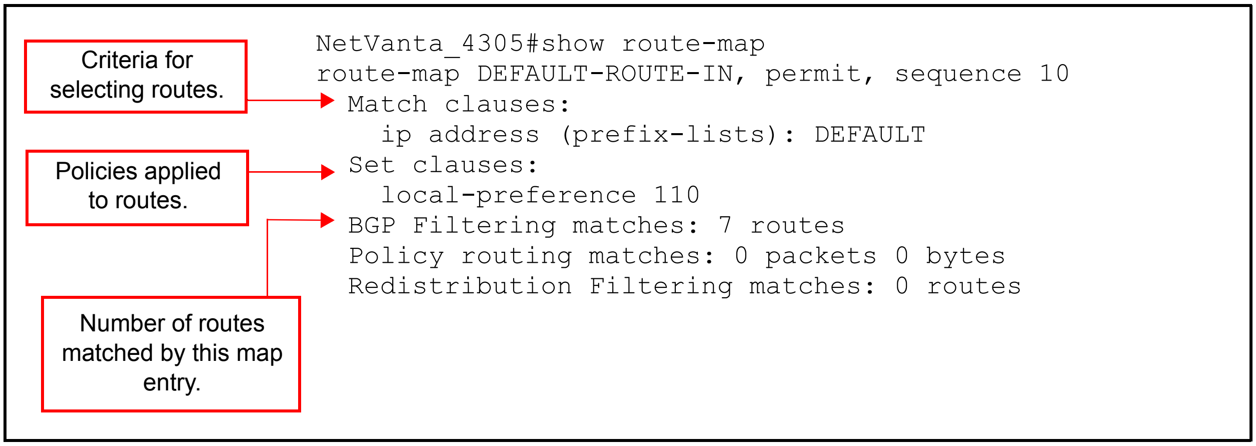
- Sequence numbers are important. The router stops processing the list after it finds a match. In *Figure 15 on page 107*, the deny statement must have a lower sequence number than the permit statement because the route specified in the deny statement also matches the permit statement.
- The **ge** and **le** keywords match prefixes equal to length specified, as well as those greater or lesser than the specified length. That is, the statement **permit 0.0.0.0/0 le 17** will allow /17 routes.

Troubleshooting a Route Map

Use the following Enable mode command to view a route map:

#show route-map [<*name*>]

Include the name of a route map to display only the route map matching the specified name. Issuing the command without entering a name will display all route maps configured on the router.



```
NetVanta_4305#show route-map
route-map DEFAULT-ROUTE-IN, permit, sequence 10
  Match clauses:
    ip address (prefix-lists): DEFAULT
  Set clauses:
    local-preference 110
  BGP Filtering matches: 7 routes
  Policy routing matches: 0 packets 0 bytes
  Redistribution Filtering matches: 0 routes
```

Criteria for selecting routes.
Policies applied to routes.
Number of routes matched by this map entry.

Figure 16. Viewing BGP Policies in a Route Map

The output shows how many routes have been matched to the route map. If the AOS device does not seem to be filtering any routes, verify that the route map has been applied to the correct neighbor and as the correct policy (inbound or outbound).

The following are useful tips to keep in mind when searching for misconfigurations in a route map:

- If attributes are to be applied to routes filtered by an inbound route map, the **set** commands must be entered for the attributes in the same route map entry in which the **match** command is entered to select permitted routes.
- If an entry does not include a match statement, then the policy in that entry will be applied to all routes.
- If an entry is being used to place a route in one or more communities with the **set community** command, then the **send-community standard** command must be issued from BGP Neighbor Configuration mode.

Other Common BGP Problems

After a BGP session has been opened and routes have been exchanged with neighbors, several problems may arise:

- A service provider may refuse to accept the local router's routes

- The local network is flooded with external traffic
- Routers are not defined in the correct communities

A Service Provider Router Refuses Local Routes

Verify that the service provider allows the advertisement of private routes.

Network Flooded with External Traffic

One of the most common uses for BGP is BGP multihoming. Multihoming allows connections to two different service providers. An unintended consequence of multihoming is that the service providers can advertise routes to each other through the multihomed router. This results in the local router becoming a transit network for external traffic.This should be prevented by utilizing prefix lists that only allow specific subnets to be advertised to each provider. This will keep the service providers from being able to advertise other routes for the Internet to each other through the local router. (For more information on proper configuration for multihoming applications, refer to *Example 3 on page 50*).

Routes in Incorrect Communities

There are several things that could prevent a remote neighbor from applying the correct policies to routes that have been defined as members of particular communities.

- The AOS device has not been enabled to send community attributes to this neighbor. Enter the **send-community standard** command from the BGP Neighbor Configuration mode context.
- The BGP neighbor defines different policies for the community or the BGP neighbor does not accept community attributes in customer routes. Consult with the service provider about what communities it supports.

There may also be problems with the local policy that has been configured for communities on the AOS device. Look at the configured route maps and examine entries that include a match clause for a community list. Then verify that the **set** clauses implement the correct policies for communities in this list.

Use the following Enable mode command to view a community list:

#show community-list <*name*>

Use the following command to view the routes that match the community list:

#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf <*name*> **ipv4 | vrf** <*name*> **ipv6]
community-list** <*name*>

Monitor the communities of routes that the router receives by entering this command:

#show bgp [any-vrf ipv4 | any-vrf ipv6 | ipv4 | ipv6 | vrf <*name*> **ipv4 | vrf** <*name*> **ipv6] community
[**<*number*>**| internet | local-as | no-advertise | no-export]**

The CLI displays all routes in the specified community. Enter the command without keywords (#**show bgp community**) for the community to see all routes known by the router that have a community attribute.

Appendix

AS Regular Expressions

Regular expressions, also known as **regexp**, are used in BGP to identify routes that are to be included in outbound routing advertisements or filtered from received inbound routing updates. Regular expressions identify an expected pattern to match against the AS path associated with a BGP route. AS paths identified by the regular expressions are subject to the actions specified in AOS.

Regular expressions are defined in the Global Configuration mode using the command **as-path-list**. Refer to *B. Filtering Routes According to AS Path on page 41* for more information on this command.

There are four types of regular expressions: atom, range, piece, and branch.

Atom

An atom is a single character.

Table 16. Atom

Character	Definition
.	Matches any single character.
^	Matches the beginning of the input string.
$	Matches the end of the input string.
\	Matches the character.

Range

A range is a sequence of characters contained within square brackets (for example, **[0-9]**).

Piece

A piece is an atom followed by one of the following symbols:

Table 17. Piece

Character	Definition
*	Matches 0 or more sequences of the atom.
+	Matches 1 or more sequences of the atom.
?	Matches the atom or the null string.

Branch

A branch is 0 or more integrated pieces.

Examples of Regular Expressions

.* Matches any character or sequence of characters.

3+ Matches any BGP route entry with at least one occurrence of the number 3 in the AS path.

\b300\b Matches any BGP route entry containing 300 in the AS path.

\b300$ Matches any BGP route entry with an AS path ending in 300.

^300\b.* Matches any BGP route entry with an AS path that begins with 300.

^$ Matches any BGP route entry with an AS path containing only the local AS.

^300\b.*301 Matches any BGP route entry with an AS path that starts with 300 and ends with 301.

^300$ Matches any BGP route entry with an AS path that contains only 300.