Chapter 27

Border Gateway Protocol version 4 (BGP-4)

Introduction

This chapter describes the Border Gateway Protocol version 4 (BGP-4), how it is implemented on the router, and how to configure the router to use it.

BGP-4 is enabled with a special feature license that you can obtain by contacting an Allied Telesis authorised distributor or reseller.

BGP-4 runs on hardware with 16 MB or more of RAM and is used with IPv4. For memory details, see the Hardware Reference for your router.

Overview of BGP-4

The Border Gateway Protocol version 4 (BGP-4) is an external gateway protocol. It allows two routers in different routing domains, known as *Autonomous Systems*, to exchange routing information to facilitate the forwarding of data across the borders of the routing domains. The basic operation of BGP-4 is described in RFC 1771, *A Border Gateway Protocol 4 (BGP-4)*.

An Autonomous System (AS) is a set of routers under a single technical administration that uses:

- one or more internal gateway protocols (IGP)
- one or more sets of common metrics to route packets within its own AS
- an external gateway protocol (EGP) to route packets to other ASs

Every public AS is identified by an Autonomous System Number (ASN) in the range 1 to 64511. An ASN in this range is a globally unique number that the IANA assigns to every AS on the internet.

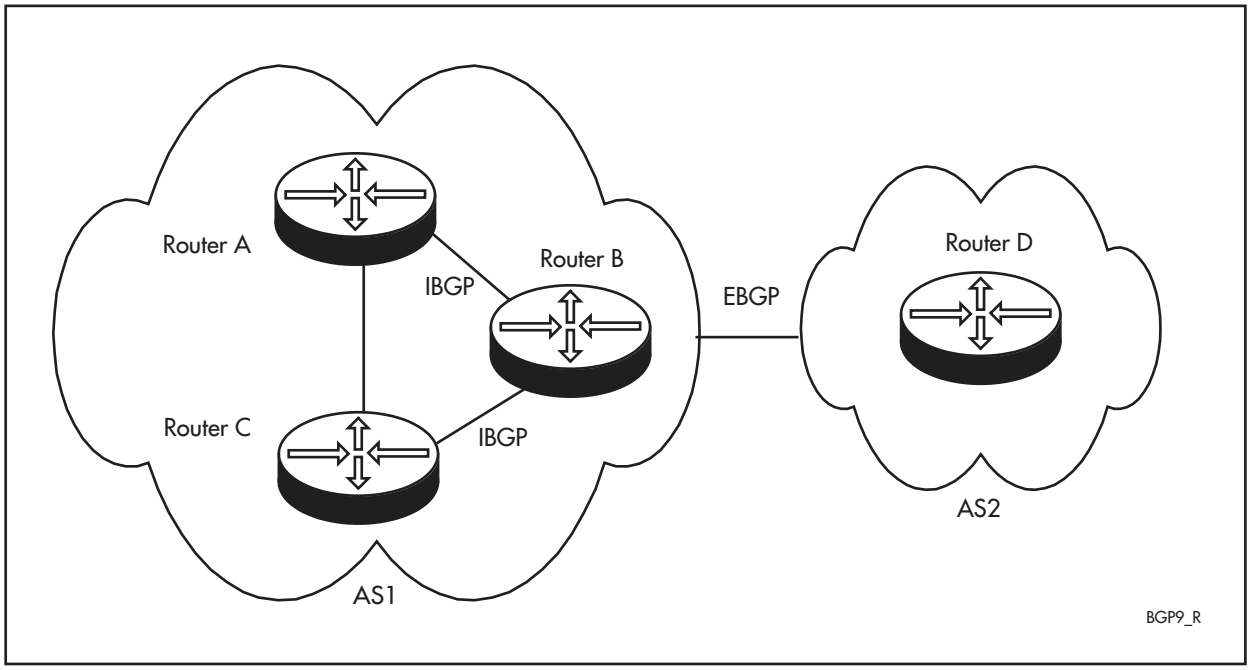
BGP lets routers learn multiple paths, choose the best path, and install it in the IP routing table. BGP-4 is based on distance vector (DV) protocol algorithms and uses TCP as its transport protocol on TCP port 179.

When BGP is used as an external gateway protocol to exchange routing information across AS borders, it is known as *External BGP (EBGP)*. EBGP connections are established between BGP *speakers* that have different AS numbers. A BGP speaker is any host that can use BGP to exchange routing information with another BGP-capable host. A BGP speaker does not necessarily have to be a router – it could be a host that passes routes learned by BGP to a router via another means, such as RIP.

A speaker may advertise any routes it knows to other speakers over an EBGP connection, as long as the speaker being advertised to has a different AS number from the speaker that is advertising.

The routes advertised over an EBGP connection can be learned by any means, for example IGP, EGP, or static assignment.

Figure 27-1: IBGP and EBGP configuration



When BGP is used as an internal gateway protocol to transfer routing information within an AS, it is known as an *Internal BGP (IBGP)*. IBGP connections are established between BGP speakers that have the same AS number. Figure 27-1 shows the use of IBGP and EBGP.

For speakers within an AS to learn routes known by all other speakers in the AS, then:

- every speaker must have IBGP connections to all the other speakers in the AS (this is called *full mesh*), or
- speakers need to be part of a confederation, or
- the AS must use route reflection. For information about route reflection, see "How to Improve IBGP Scalability" on page 27-33.

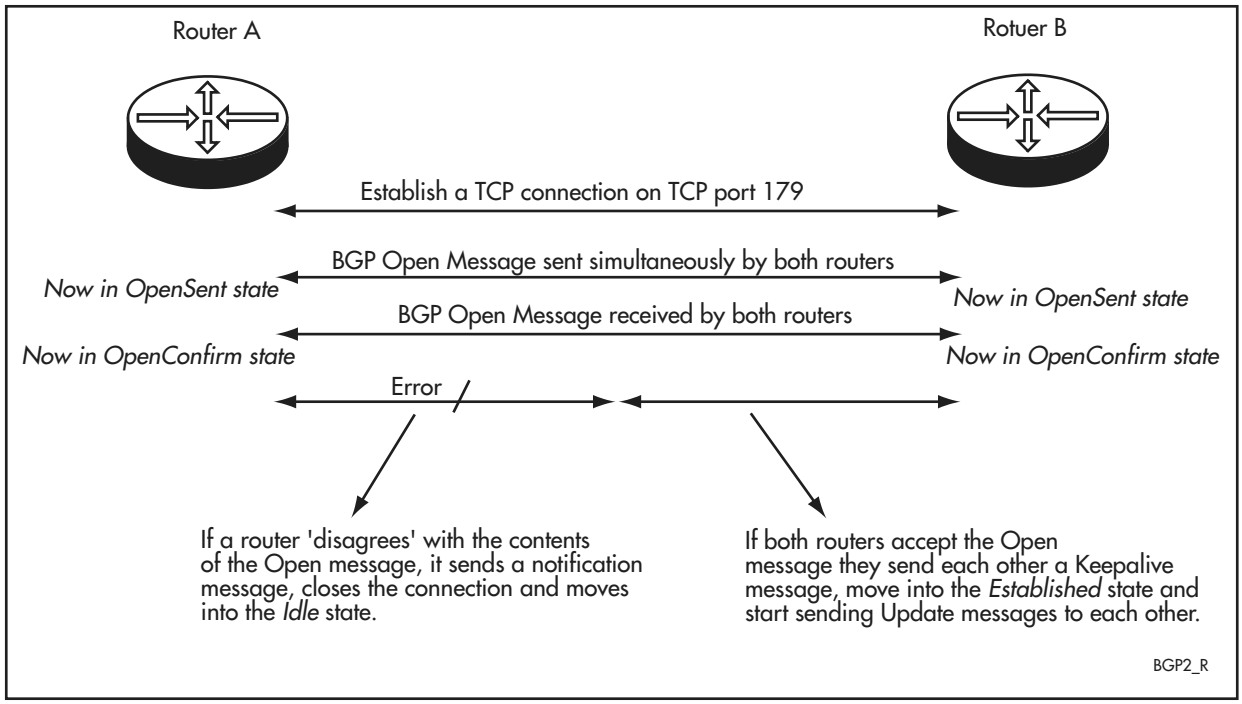
Without route reflection, a speaker cannot advertise routes learned over an IBGP connection to another speaker over another IBGP connection. However, a speaker can advertise routes learned from other means (for example, RIP and OSPF) to another speaker over an IBGP connection. A speaker can also advertise routes learned from an IBGP connection to another speaker over an EBGP connection.

An AS with one BGP speaker and a single external BGP connection is referred to as a *stub* AS.

BGP Operation

BGP is a protocol between two BGP speakers, which are called *peers*. Two routers become BGP peers when a TCP connection is established on Port 179 between them. The communication flow is illustrated in Figure 27-2.

Figure 27-2: Communication flow in a BGP session



Establishing a connection

BGP peer sessions start in the Idle state. In this state, BGP refuses all incoming BGP connections and does not allocate resources to peers. When you trigger a Start event—by enabling a peer—the router initiates a TCP connection to the peer and moves that peer session into the Connect state.

If the TCP connection attempt to a peer fails, the session moves into the Active state, waits until its ConnectRetry time expires, then tries to establish the connection again.

When the TCP connection is established, BGP peers immediately identify themselves to each other by simultaneously sending *open* messages, and move into the OpenSent state. The open messages let the peers agree on various protocol parameters, such as timers, and negotiate shared capabilities.

When each router receives an open message, it checks all the fields. If it "disagrees" with the contents of the open message, it sends a *notification* message, closes the connection and goes into the Idle state. If it finds no errors, it moves into the OpenConfirm state and sends back a *keepalive* message.

When both routers have received a keepalive message, they move into the Established state. The BGP session is now open. BGP sessions typically stay in the Established state most of the time. They only leave the Established state if an error occurs, or the hold time expires with no contact from the far end.

Exchanging routing information

Once a router has established a BGP connection with a peer, it starts to exchange routing information with that peer. Initially the peers exchange a complete copy of their routing tables. After this, they exchange updates to this routing information, in update messages.

Routing information contained within an update message consists of:

- a set of attribute values

  Attributes describe properties of the routes the update message contains. They are described in detail in "BGP Attributes" below.

- a list of one or more prefixes

  A prefix is the network address and CIDR mask. Each prefix contained within an update message represents a network that can be reached through the IP address given in the NextHop attribute contained in the same update message.

  The attribute values contained in the attributes section of the update message apply to *all* the prefixes that are advertised in that update message. Update messages can advertise multiple routes by listing multiple prefixes, as long as all their attributes are the same.

Update messages can also list routes that are withdrawn from service. These routes do not have to have the same attributes as the advertised routes.

Maintaining and closing a connection

Peers regularly exchange keepalive messages to prevent sessions from expiring. These messages are sent every 1 to 21845 seconds, every 30 seconds by default.

When an error occurs during a BGP session, the router that perceives the error sends a notification message to its peer identifying the error. It then closes the TCP connection and moves into the Idle state. Each router stops using routing information it heard from the other.

If a BGP speaker receives neither an update message or a keepalive message from a peer for a configurable period of time, called the hold time, it resets the session and withdraws any routes it learned from that peer.

BGP Attributes

An important part of the BGP protocol operation is the set of *attributes* associated with prefixes. Each BGP update message contains a set of attributes (Table 27-1 on page 27-7). These attributes describe some of the properties of the routes, and can be used in making decisions about which routes to accept and which to reject, in the following ways:

- If the router has multiple routes to a destination, it checks the attributes of each route to determine which one to use (see "BGP Route Selection" on page 27-8).
- You can create filters to reject or accept routes on the basis of their attributes (see Chapter 28, Filtering IP Routes).
- You can create route maps to change the attributes of particular update messages (see "Creating Route Maps for BGP Routes" on page 28-10 of Chapter 28, Filtering IP Routes).

Table 27-1: BGP attributes

Attribute	Description
Origin	How the prefix came to be routed by BGP at the origin AS. The router can learn prefixes from various sources and then put them into BGP. Sources include directly connected interfaces, manually configured static routes, and dynamic internal or external routing protocols. Values are IGP (internal protocols such as RIP and OSPF), EGP (other EGPs) and INCOMPLETE (static routes or other means). Every update message has this attribute.
AS_path	A list of the autonomous systems through which the announcement for the prefix has passed. As prefixes pass between autonomous systems, each autonomous system adds its Autonomous System Number (ASN) to the beginning of the list. This means the AS_path can be used to make routing decisions. Every update message has this attribute, although it may be empty.
Next_hop	The address of the next node to which the router should send packets to get the packets closer to the destination. Every update message has this attribute.
Multi_Exit_Discriminator (MED)	A metric expressing the optimal path by which to reach a particular prefix in or behind a particular AS. One AS sets the value and a different AS uses that value when deciding which path to choose.
Local_preference	A metric used in IBGP so each host knows which path inside the AS it should use to reach the advertised prefix. EBGP peers do not send this value, and ignore it on receipt.
Atomic_aggregate	An attribute that allows BGP peers to inform each other about decisions they have made about overlapping routes. If Router A receives overlapping routes, and selects the less specific (more general route) only, then it attaches the atomic_aggregate attribute. When one of its neighbours receives a prefix with the atomic_aggregate attribute set, that neighbour must not take the prefix and de-aggregate it into any more specific entries in BGP.
Aggregator	An attribute that can be attached to an aggregated prefix to specify the AS and IP address of the router that performed the aggregation.
Community	Where the prefix is relevant and should be advertised. By default, all prefixes belong to the Internet community, which is the community of all BGP peers. Other communities have been globally defined that limit the scope of prefix advertisement or export, or you can identify a community by a community number.
Originator_ID	The router ID of the IBGP peer that first learned this route, either via an EBGP peer or by some other means such as importing it. This attribute is used by route reflection to prevent routing loops. EBGP peers do not send this value, and ignore it on receipt.
Cluster_list	A list of the cluster IDs of route reflectors who have reflected the corresponding route(s) within this AS. This attribute is used by route reflection to prevent routing loops. EBGP peers do not send this value, and ignore it on receipt.

BGP Route Selection

The route selection process involves selecting the best route towards a prefix from all the routes that exist in the BGP RIB. BGP can select from all the routes that it has learned and accepted except for any routes that are unreachable, such as routes that are withdrawn or damped. When BGP selects a route as the best for a particular prefix, it adds this route into the IP routing table, and advertises the route to all appropriate neighbouring peers.

When there is only one route toward a particular prefix, that route is selected as the best route. When there are multiple routes toward a particular prefix, then BGP uses the rules in the following table to decide which one to select. If a rule results in selection of a single route, the router uses this route. If multiple routes match a rule, the router goes to the next rule. Rule 10 guarantees that BGP can select only one route, as it is not possible to have multiple different peer sessions that share the same IP address.

If BGP detects that a selected route is no longer the best route, BGP sends withdraw messages for that route to its neighbouring peers. These withdrawal messages can cause the peers to propagate the withdrawal to other devices. If the router and its peers repeatedly send withdrawals, this can causing route flapping on the network. You can use BGP route flap damping to prevent this scenario from occurring. For further information about route flapping and BGP route flap damping, see "How to Minimise the Impact of Unstable EBGP Routes" on page 27-24.

Rule	For this...	the router chooses the route that...
1	local_preference	has the highest local preference. How the router determines the local preference depends on the source of the route: • For routes the router learned via an EBGP session, or for routes it learned from sources such as an IGP or static configuration, the router calculates the value of the preference itself. • For routes the router learned from an IBGP peer, the router uses the preference supplied by the peer—the update message for that route contains a local_preference attribute indicating the degree of preference.
2	route type	came into the BGP routing table from a preferred source. The order of preference is: a. routes imported into the BGP routing table from the router's RIB, using BGP import or network entries b. routes learned through a BGP aggregate entry c. routes learned from a foreign peer of any type, such as an EBGP, IBGP or confederation peer
3	AS_path	has the shortest AS path.
4	origin	has the preferred origin. The order of preference is: a. IGP b. EGP c. INCOMPLETE

Rule	For this...	**the router chooses the route that...** (cont)
5	Multi_Exit_Discriminator value	has the lowest MED value. This rule applies if the local system is configured to take into account the value of the Multi_Exit_Discriminator (MED), and if the multiple routes are learned from the same neighbouring AS.
6	connection type	it learned over an EBGP connection. BGP considers that it has learned a route over an IBGP connection if the route contains AS confederation sets or sequences in its AS path. Note that candidate routes' AS paths only contain EBGP confederation AS numbers, because BGP drops routes with the local AS path in their path list.
7	next_hop attribute	has the minimum cost to the next hop specified in the next_hop attribute. Deciding the cost involves looking into the IP route table.
8	router ID	it learned from the peer with the lowest router ID. The peer's router ID is determined by the following rules: • If the peer has been configured with a router ID by using the command **set bgp routerid=*ipadd***, that address is used as its router ID. • Otherwise, if a local IP address has been set for the peer, that address is used as its router ID. • Otherwise, if neither has been set, the highest IP address configured on any of the peer's interfaces is used as its router ID.
9	cluster list	has the shortest cluster list. The cluster list attribute only exists within Autonomous Systems that use route reflection, so if the router's AS does not use route reflection, the cluster list is treated as having a length of zero.
10	neighbour address	it learned from the peer with the lowest neighbour IP address. The neighbour IP address is the address that the peer uses for the TCP connection that supports the peer session. For more information about the address routers use, see "How to Set the IP Address that Identifies the Router" on page 27-40.

Classless Inter-domain Routing (CIDR) and Aggregation

Interfaces, static routes, and routes learned via IGP (such as RIP and OSPF) can have a specific classful subnet mask (Class A, B, C or D). However, BGP routes are *classless*, (whereby IP addresses are not part of a class) so that shorter route tables can be exchanged and the number of advertised routes (prefixes) is fewer.

Figure 27-3 shows an example of inter-domain routing without CIDR. The service provider has many customers, all with Class C addresses that start with 204.71. The service provider announces each of the networks individually into the global Internet routing mesh.

Figure 27-3: Inter-domain routing without CIDR

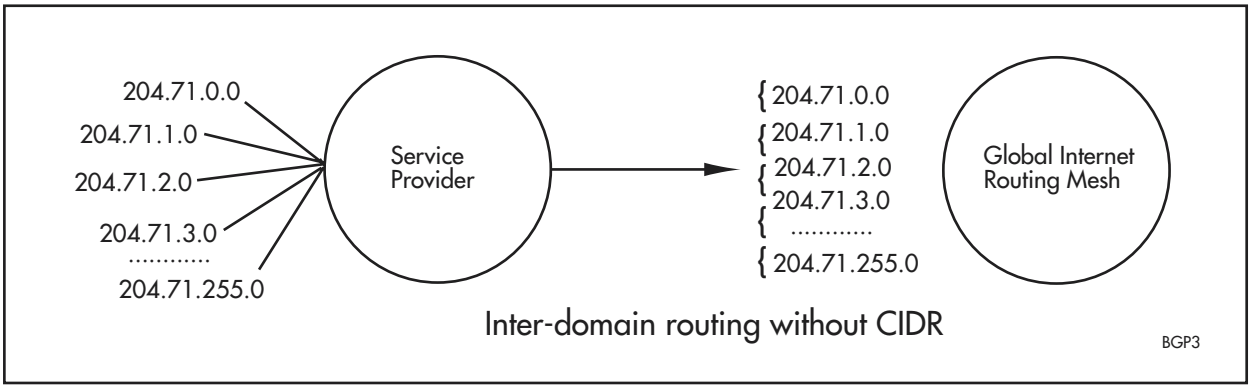
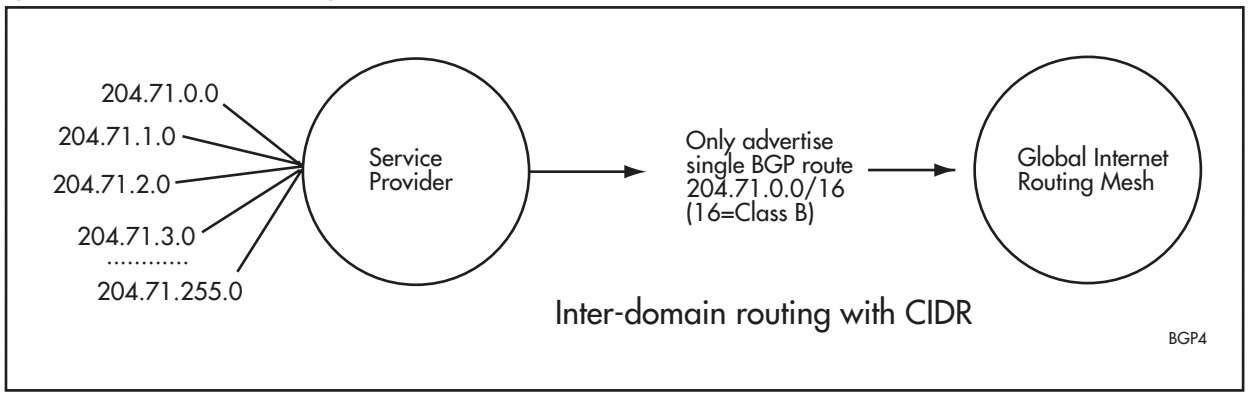


Figure 27-4 shows an example of inter-domain routing with CIDR. By routing with CIDR, the service provider can aggregate the classful networks used by its customers into single classless advertisements. This provides routing for hundreds of customers by announcing only one advertisement into the global Internet routing mesh.

Figure 27-4: Inter-domain routing with CIDR



CIDR also copes with overlapping routes/prefixes because the route with the longest match (greatest number of bits/more specific netmask) is always chosen first. As the prefix traverses Autonomous Systems, each AS adds its AS number to the AS path bits, in both the BGP attribute and also the source IP and next-hop address.

BGP Multi-Homing

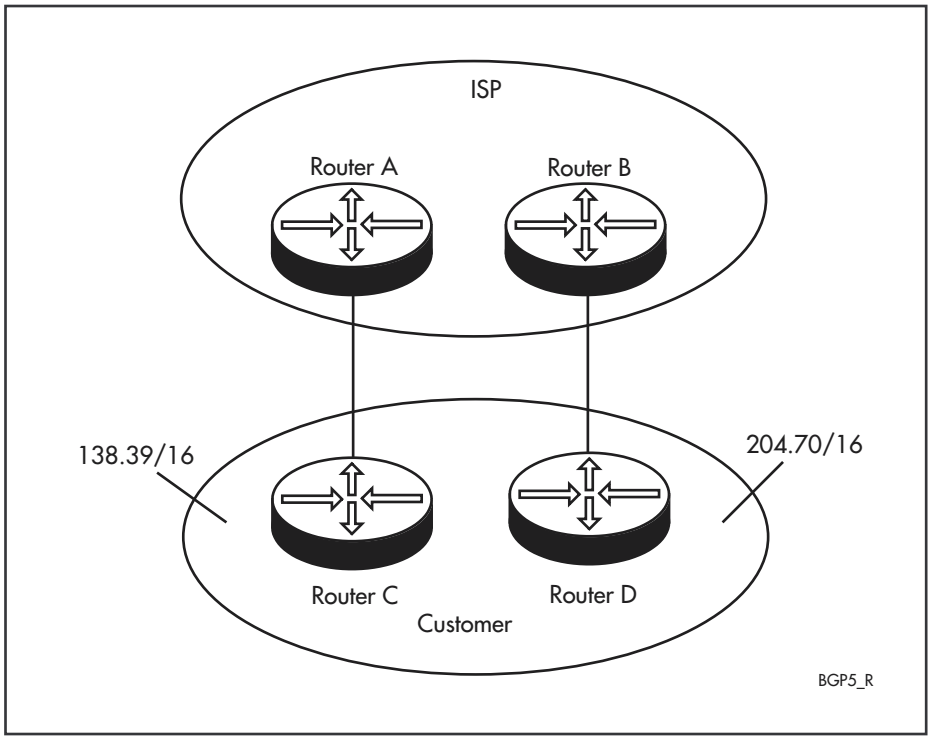
An AS may have multiple EBGP speakers connected to different ASs. This is known as BGP *multi-homing*. Multi-homing improves reliability of the AS connection to the Internet, and improves network performance because the network's bandwidth is the sum of all the circuits' bandwidth. Network performance increases when more than one connection is used at a time; otherwise, maximum performance is the bandwidth being used by one connection at a given time. An even split of traffic across multiple connections is called *load balancing*.

Sites can be multi-homed in the following ways:

- to a single Internet Service Provider (ISP) or Network Service Provider (NSP)
- to more than one ISP or NSP

The following figure illustrates the most reliable multi-homing topology to a single ISP involving different routers in the ISP and different routers in the customer network.

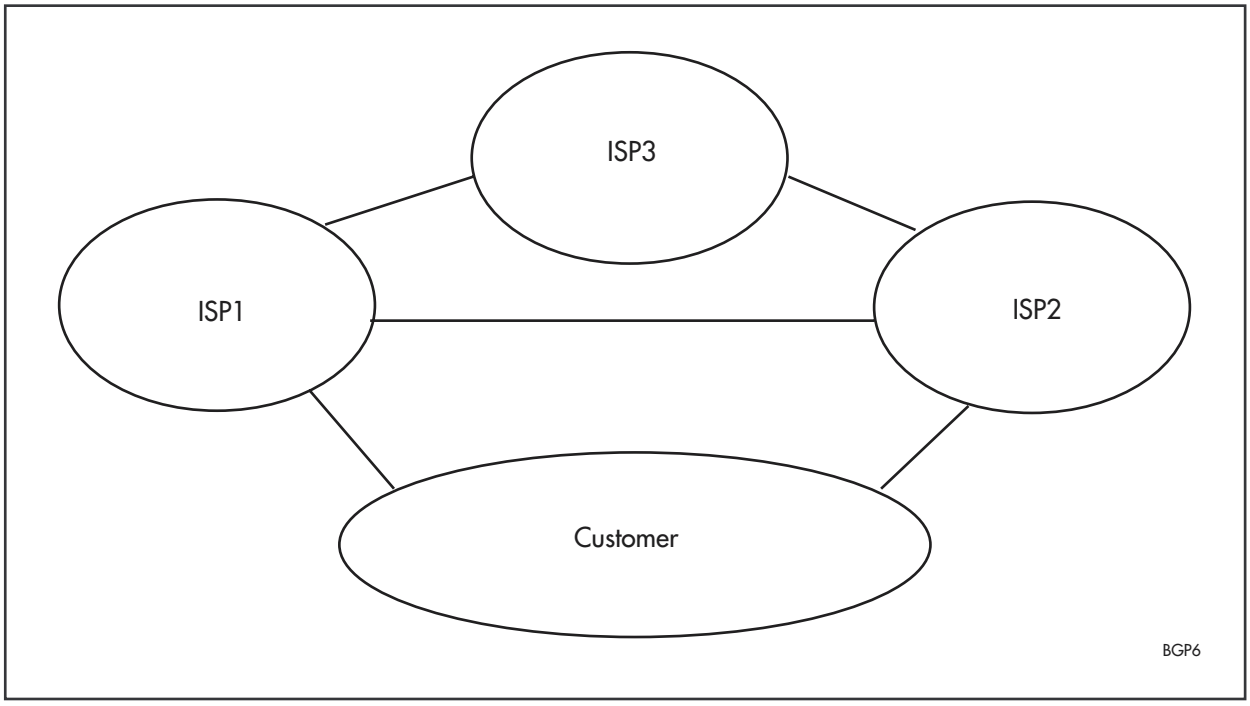
Figure 27-5: Multi-homing to a single ISP



This example is the most reliable because no equipment is shared between the two links. If the traffic between the two networks is equal, the approach to load balancing would be to use the link between Router A and Router C for traffic going to 138.39/16 and use the link between Router B and Router D for traffic going to 204.70/16.

Multi-homing to more than one provider is shown in Figure 27-6. The customer is multi-homed to ISP1 and ISP2; ISP1, ISP2, and ISP3 connect to each other. The customer has to decide how to use address space, as this is critical for load balancing from the ISPs to the customer, whether it delegates it by ISP1, ISP2, both, or independently.

Figure 27-6: Multi-homing to more than one provider



When the customer uses the address space delegated to it by ISP1, the customer uses a more specific prefix out of ISP1's aggregate and ISP1 can announce only the aggregate to ISP2. When the customer gets as much traffic from ISP1 as it gets from both ISP2 and ISP3, load balancing can be good. When ISP2 and ISP3 together send substantially more traffic than ISP1, load balancing can be poor. When the customer uses the address space delegated to it by ISP2, it does the same although ISP1 is the ISP to announce the more-specific route and attract traffic to it. Load balancing may be quite good if ISP1's address space is used, but not very good if ISP2's space is used.

When the customer uses the address space delegated to it by both ISP1 and ISP2, the degree of load balancing from ISP1 and ISP2 to the customer depends on the amount of traffic destined for the two ISPs. If the amount of traffic for the two is about the same, load balancing towards the customer can be quite good, if not, load balancing can be poor.

The option of the customer getting its own address space from a registry rather than from either ISP1 or ISP2 provides the most control but there is no aggregation. *Aggregation* is the combining of several different routes so that a single route can be advertised. This minimises the size of the routing table. The customer needs address space that makes it through the *route filters* (see "BGP Route Filtering"); otherwise, the customer may end up having no connectivity. If the customer gets address space that makes it through the route filters, there must be control over which provider uses which path to reach the customer.

When ISP1 is the largest, it may want to reach the customer via the ISP1-customer link, but have ISP2 and ISP3 go through the ISP2-customer link. When the path learned from ISP2 is shorter than the path learned from ISP1, ISP3 uses ISP2 to reach the customer.

BGP Route Filtering

BGP route filtering enables you to filter the routing information that your routers receive from the networks they connect to, and that they advertise to those networks. This gives you control over the path of any traffic originating from or traversing your network.

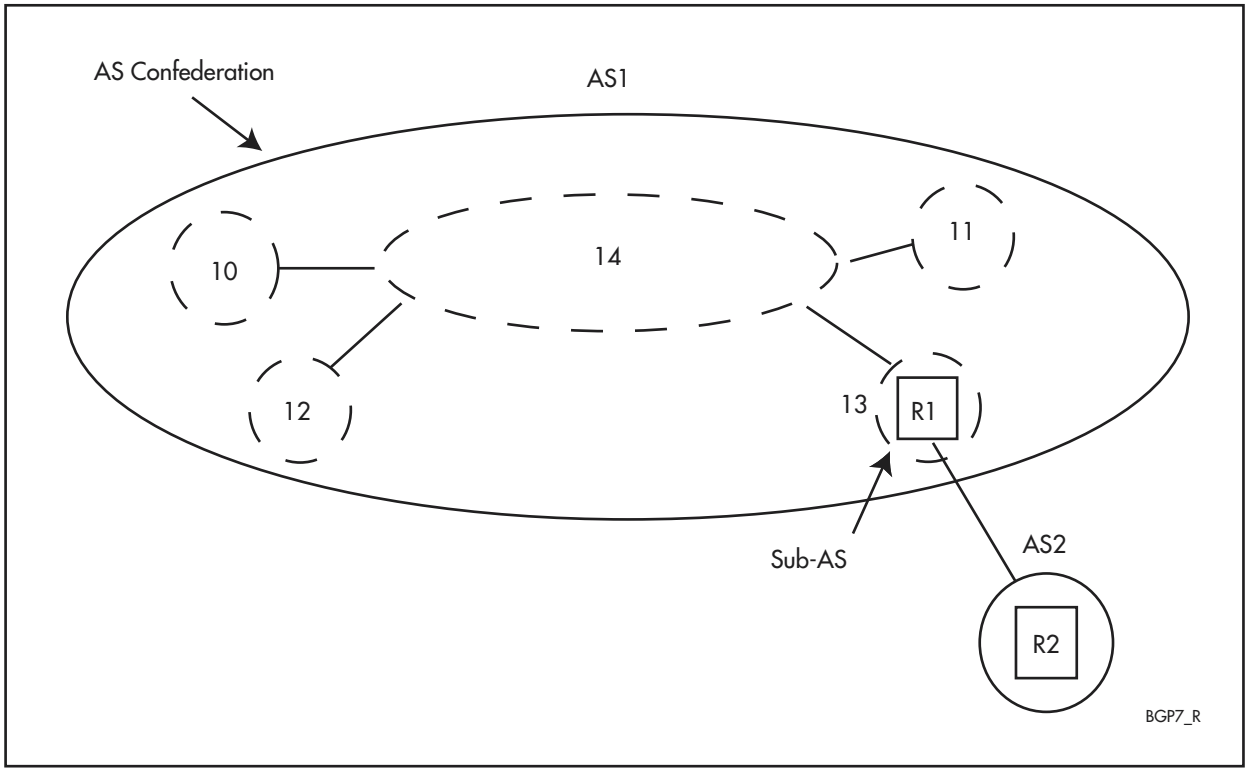
Route filtering is described in detail in Chapter 28, Filtering IP Routes.

AS Confederations

An AS confederation is a collection of autonomous systems that are advertised as a single AS number to BGP speakers that are not members of the confederation. The autonomous systems in a confederation communicate between themselves using Confederation BGP (CBGP). AS confederations are used to subdivide autonomous systems with a very large number of BGP speakers to control routing policy.

Figure 27-7 is an example of an AS confederation. AS1 has been split into several sub-ASs (AS10-AS14). The original AS does not look any different to router 2, which is outside the confederation. Router 2 still sees AS1 rather than AS10-AS14. This implies that routers within the confederation are configured with an AS number for the confederation (for example, AS1), and an *AS member number*, an AS number visible only to those members within the confederation (for example, AS10).

Figure 27-7: Example of an AS confederation



Splitting a large AS into several smaller ones significantly reduces the number of intra-domain BGP connections. Unfortunately, splitting an AS may increase the complexity of routing policy based on AS path information for all members on the Internet. It also increases the maintenance overhead of coordinating external peering when the internal topology of this collection of ASs is modified.

Dividing an AS may unnecessarily increase the length of the sequence portions of the AS path attribute, and may adversely affect optimal routing of packets through the Internet.

Triggers

The trigger facility automatically runs specific command scripts when particular triggers are activated. When an event activates a trigger, parameters specific to the event are passed to the script that is run. For a full description of the trigger facility, see Chapter 59, Trigger Facility.

Triggers can be created for the following BGP events:

- when the router runs low on memory
- when a peer changes state

Module MODULE=BGP

Event MEMORY
Description The router has run low enough on memory that BGP has had to start dropping routes.
Parameters There are no command parameters for this event.
Script Arguments There are no arguments to pass to the script.

Event PEERSTATE
Description The PEERSTATE trigger causes a trigger whenever a state change occurs that matches the peer, state and direction conditions. If the state is ANY and the direction is BOTH, two triggers are generated, one for leaving the old state and one for entering the new state.
Parameters The following command parameters can be specified in the **create** and **set trigger** commands:

Parameter	Description
peer=any|<*ipaddress*>	The IP address of the peer for which the state changes are interested in. This parameter is required in the **create trigger** command for BGP triggers, but is optional in the **set trigger** command unless the trigger event is changing from **memory** to **peerstate**.
bgpstate=idle|connect|active|opensent|openconfirm|established|any	The BGP state for which the trigger is required. This parameter is required in the **create trigger** command for BGP triggers, but is optional in the **set trigger** command, unless the trigger event is changing from **memory** to **peerstate**.
direction=enter|leave|both	Whether a match is made for the state the peer is leaving, entering, or both. This parameter is required in the **create trigger** command for BGP triggers, but is optional in the **set trigger** command unless the trigger event is changing from **memory** to **peerstate**.

Script Arguments The trigger passes the following arguments to the script:

Argument	Description
%1	The peer ID of the peer that has just undergone the state change.
%2	The state just left or entered.
%3	Whether the state was left or entered.

Example To create trigger 1, which activates whenever the router becomes low on memory, initiating the script MEMLOW.SCP, use the command:

```
create trigger=1 module=bgp event=memory script=memlow.scp
   repeat=yes
```

To create trigger 2, which activates whenever peer=172.30.1.2 leaves the ESTABLISHED state, initiating the script PEERDOWN.SCP, use the command:

```
create trigger=2 module=bgp event=peerstate peer=172.30.1.2
   bgpstate=established direction=leave script=peerdown.scp
   repeat=yes
```

To modify trigger 2, which activates when any peer leaves the ESTABLISHED state, use the command:

```
set trigger=2 peer=any
```

Redistributing BGP Routes

You can redistribute BGP routes into the following protocols:

- OSPF—see "Importing BGP routes into OSPF" on page 26-15 of Chapter 26, Open Shortest Path First (OSPF)
- RIP—see "BGP Routes" on page 25-4 of Chapter 25, Routing Information Protocol (RIP)

Configuring BGP Peers

This section describes basic BGP configuration:

- **How to Create a Basic BGP AS**
- **How to Create BGP Peers Using Peer Templates**
- **How to Modify BGP Peers (Without Templates)**
- **How to Use a Template to Modify BGP Peers**
- **How to Modify BGP Peers that Use a Template**
- **How to Delete BGP Peers**

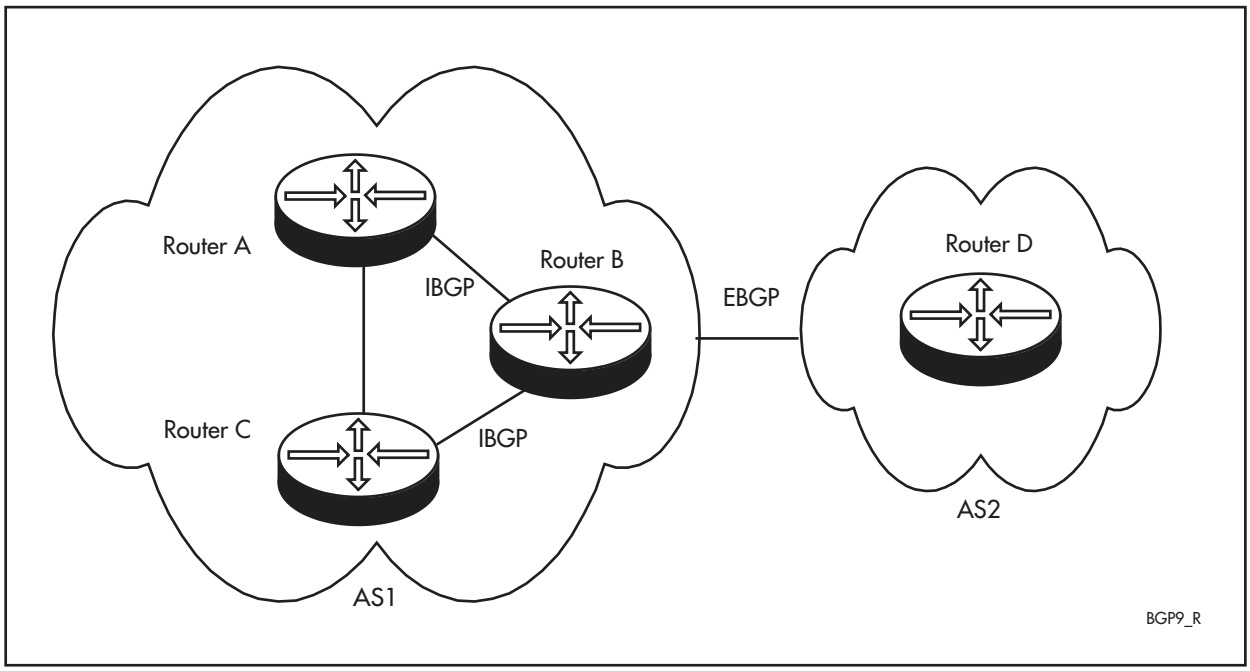
How to Create a Basic BGP AS

This section describes how to configure your network as an Autonomous System by using EBGP to send and receive routing information from an external peer (for example, an ISP), and by using IBGP to communicate routes within the AS.

For this basic BGP setup, you need to configure:

- the external BGP speaker. This is the router connected to the remote peer. In Figure 27-8, router B is the external speaker and router D is the remote peer. For the configuration procedure, see Table 27-2 on page 27-17. For checking and debugging, see Table 27-4 on page 27-19.
- the internal BGP speakers. These are all the other routers in the AS, connected to the external BGP speaker. In Figure 27-8, routers A and C are internal speakers. For the configuration procedure, see Table 27-3 on page 27-18. For checking and debugging, see Table 27-4 on page 27-19.

Figure 27-8: Example of the use of IBGP and EBGP



External speakers

Table 27-2: Procedure for configuring the external BGP speaker

Step	Command	Action
1		Configure the lower-layer protocols that link the router to the remote peer, for example frame relay, PPP.
2	**set ip autonomous**=1..65534	Assign your AS number to the router.
3	**add ip interface**=*interface* ipaddress=*ipadd* [*other-options*] **add ip route**=0.0.0.0 interface=*interface* nexthop=*remote-peer-ipadd* **enable ip**	Configure IP on the interface that links the router to the remote peer: • assign an IP address • create a default route, if the IP addresses assigned to the interfaces connecting routers B and D are on different subnets • enable IP
4	**add ip interface**=*interface* ipaddress=*ipadd* [*other-options*]	Configure IP on the interfaces that link the router to each internal speaker.
5	**set bgp** routerid=*ipadd* [*other-options*]	Configure an interface on the router to be the source of IP packets generated by BGP. This step is not required but is suggested as good practice.
6	**add bgp peer**=*remote-peer-ipadd* remoteas=*remote-peer-asn* [authentication={md5|none}] [client={yes|no}] [connectretry={default|0..4294967295}] [description={none|*description*}] [ehops={default|1..255}] [fastfallover={no|yes}] [holdtime={default|0|3..65535}] [infilter={none|*prefixlist-name*}] [inpathfilter={none|1..99}] [inroutemap={none|*routemap*}] [keepalive={default|1..21845}] [local={none|1..15}] [maxprefix={off|1..4294967295}] [maxprefixaction={terminate|warning}] [minasoriginated={default|0..3600}] [minrouteadvert={default|0..3600}] [nexthopself={no|yes}] [outfilter={none|*prefixlist-name*}] [outpathfilter={none|1..99}] [outroutemap={none|*routemap*}] [password=*password*] [privateasfilter={no|yes}] [sendcommunity={no|yes}]	Add the remote peer to the router. For the **peer** parameter, enter the IP address of the interface, on the remote peer, to which the external speaker connects. For the **remoteas** parameter, enter the remote peer's ASN.
7	**enable bgp peer**=*remote-peer-ipadd*	Enable the peer. The router establishes a BGP connection with the remote peer and exchanges routing information.

Internal speakers

Table 27-3: Procedure for configuring an internal BGP speaker

Step	Command	Action
1		Connect the router directly to the external speaker and configure lower-layer protocols as required, for example VLANs.
2	**set ip autonomous**=1..65534	Assign your AS number to the router.
3	**add ip interface**=*interface* ipaddress=*ipadd* [*other-options*] **add ip route**=*ext-speaker-ipadd* interface=*interface* nexthop=*ipadd* **enable ip**	Configure IP on the interface that links the router to the external speaker: • assign an IP address • create a route to the external speaker if necessary • enable IP
4	**set bgp** routerid=*ipadd* [*other-options*]	Configure an interface on the router to be the source of IP packets generated by BGP. This step is not required but is suggested as good practice.
5	**add bgp peer**=*external-speaker-ipadd* remoteas=*external-speaker-asn* [authentication={md5|none}] [client={yes|no}] [connectretry={default|0..4294967295}] [description={none|*description*}] [ehops={default|1..255}] [fastfallover={no|yes}] [holdtime={default|0|3..65535}] [infilter={none|*prefixlist-name*}] [inpathfilter={none|1..99}] [inroutemap={none|*routemap*}] [keepalive={default|1..21845}] [local={none|1..15}] [maxprefix={off|1..4294967295}] [maxprefixaction={terminate|warning}] [minasoriginated={default|0..3600}] [minrouteadvert={default|0..3600}] [nexthopself={no|yes}] [outfilter={none|*prefixlist-name*}] [outpathfilter={none|1..99}] [outroutemap={none|*routemap*}] [password=*password*] [privateasfilter={no|yes}] [sendcommunity={no|yes}]	Add the external speaker to the router as a BGP peer. For the **peer** parameter, enter the IP address of the external speaker's interface to which this internal speaker connects. For the **remoteas** parameter, enter the external speaker's ASN (which is the same as the internal speaker's ASN).
6	**enable bgp peer**=*ext-speaker-ipadd*	Enable the peer. The router establishes a BGP connection with the external speaker and exchanges routes.

Checking peers

Table 27-4: Procedure for checking and debugging BGP peers

Step	Command	Action
1	**show bgp peer** **show bgp peer**=*external-speaker-ipadd*	Check that the connections are established and that the internal and external speakers are exchanging messages. For example output and definitions, see Figure 27-36 on page 27-125 and Figure 27-37 on page 27-126.
2	**show bgp**	Check the number of routes learned, and other information. For example output and definitions, see Figure 27-20 on page 27-107.
3	**show bgp route**[=*prefix*] [community={internet|noadvertise| noexport|noexportsubconfed|*aa*:*xx*}[,...]}] [peer=*ipadd*] [regexp=*aspathregexp*]	List information about all or a subset of the learned routes. Note that BGP may learn many thousands of routes. For example output and definitions, see Figure 27-39 on page 27-133.
4	**enable bgp** **debug**={damping|msg|state|update|all} [,...] [peer=*ipadd*]	Enable BGP debugging. Note that debugging may produce very large amounts of data.

How to Create BGP Peers Using Peer Templates

Peer templates make it easier to create BGP peers when many peers have identical inbound and outbound filtering policies, or timer values. They enable you to define a template set of these values, which you can subsequently apply to many different peers. You can assign a template to a BGP peer either when you create the peer, or afterwards.

Table 27-5: Procedure for using a template to create a BGP peer

Step	Command	Action
1	**add bgp peertemplate**=1..30 [client={yes|no}] [connectretry={default|0..4294967295}] [description={none|*description*}] [holdtime={default|0|3..65535}] [infilter={none|*prefixlist-name*}] [inpathfilter={none|1..99}] [inroutemap={none|*routemap*}] [keepalive={default|1..21845}] [local={none|1..15}] [maxprefix={off|1..4294967295}] [maxprefixaction={terminate|warning}] [minasoriginated={default|0..3600}] [minrouteadvert={default|0..3600}] [nexthopself={no|yes}] [outfilter={none|*prefixlist-name*}] [outpathfilter={none|1..99}] [outroutemap={none|*routemap*}] [privateasfilter={no|yes}] [sendcommunity={no|yes}]	Create the template. You can specify most of the peer settings in the template.
2	**show bgp peertemplate**[=1..30]	Check the template settings.
3	**add bgp peer**=*ipadd* remoteas=*asn* policytemplate=1..30 [authentication={md5|none}] [password=*password*] [description={none|*description*}] [ehops={default|1..255}] [fastfallover={no|yes}]	Create the peer entry and attach the template to it. You can also specify peer settings that are not available in the template.

How to Modify BGP Peers (Without Templates)

To modify a peer, unless the peer is using a template, use the command:

```
set bgp peer=ipadd [other-options]
```

You do not need to disable the peer first.

For information on changing peers that use templates, see "How to Modify BGP Peers that Use a Template" .

Once you have modified the peer, the router needs to update that peer. The router supports the following RFCs for updating modified peers:

- RFC 2918, *Route Refresh Capability for BGP-4*
- RFC 2842, *Capabilities Advertisement with BGP-4*

Automatic updates

You can configure the router to make updates automatically by using the command:

```
enable bgp autosoftupdate
```

This is disabled by default. Note that you must enable automatic updating before you modify the peer.

Manually-triggered updates

Alternatively, you can manually trigger the BGP peer to reset by using the command:

```
reset bgp peer={all|ipadd} soft={in|out|all}
```

The **soft** parameter determines the direction to update. There are two types of updates:

- Inbound updates, which reset routes that the router receives from the peer. To trigger these, the router sends a Route Refresh message to the peers from which it receives routes. The Route Refresh message triggers the peers to resend a BGP Update message.
- Outbound updates, which reset routes the router sends. To reset these, the router simply sends a BGP Update message to the affected BGP peers.

If you do not manually or automatically trigger an immediate update, changes to the peer take effect when the router next receives an update message from that peer or sends an update message to it.

To see if automatic updating is enabled, use one of the commands:

```
show bgp
```

```
show bgp peer
```

In the command **show bgp peer**, you can see that the router and its peer have negotiated automatic updating when the "Capabilities" entry contains "Route Refresh". This command also displays the number of route refresh messages received from and sent to the peer.

How to Use a Template to Modify BGP Peers

You can apply a template to an existing peer, which overrides its current settings with the template settings.

Table 27-6: Procedure for using a template to modify a BGP peer

Step	Command	Action
1	**add bgp peertemplate**=1..30 [client={yes|no}] [connectretry={default|0..4294967295}] [description={none|*description*}] [holdtime={default|0|3..65535}] [infilter={none|*prefixlist-name*}] [inpathfilter={none|1..99}] [inroutemap={none|*routemap*}] [keepalive={default|1..21845}] [local={none|1..15}] [maxprefix={off|1..4294967295}] [maxprefixaction={terminate|warning}] [minasoriginated={default|0..3600}] [minrouteadvert={default|0..3600}] [nexthopself={no|yes}] [outfilter={none|*prefixlist-name*}] [outpathfilter={none|1..99}] [outroutemap={none|*routemap*}] [privateasfilter={no|yes}] [sendcommunity={no|yes}]	Create the template. You can specify most of the peer settings in the template.
2	**show bgp peertemplate**[=1..30]	Check the template settings.
3	**set bgp peer**=*ipadd* remoteas=*asn* policytemplate=1..30 [authentication={md5|none}] [password=*password*] [description={none|*description*}] [ehops={default|1..255}] [fastfallover={no|yes}]	Attach the template to the peer. You can also specify peer settings that are not available in the template.
4	**reset bgp peer soft** reset bgp peer=*ipadd* soft={in|out|all}	If automatic updating is not enabled, trigger the peer to update.

How to Modify BGP Peers that Use a Template

Once you have assigned a template to a peer, the method you use to modify the peer depends on the type and scope of the modification.

- To change a parameter on **all** peers that use the template, when the parameter **is** available in the template, change the template. Use the command:

```
set bgp peertemplate=1..30 [client={yes|no}]
    [connectretry={default|0..4294967295}]
    [description={none|description}]
    [holdtime={default|0|3..65535}]
    [infilter={none|prefixlist-name}]
    [inpathfilter={none|1..99}]
    [inroutemap={none|routemap}]
    [keepalive={default|1..21845}] [local={none|1..15}]
    [maxprefix={off|1..4294967295}]
    [maxprefixaction={terminate|warning}]
    [minasoriginated={default|0..3600}]
    [minrouteadvert={default|0..3600}]
    [nexthopself={no|yes}]
    [outfilter={none|prefixlist-name}]
    [outpathfilter={none|1..99}]
    [outroutemap={none|routemap}]
    [privateasfilter={no|yes}] [sendcommunity={no|yes}]
```

- To change an **individual** peer when the parameter **is not** available in the template, specify the peer and parameter by using the command:

```
set bgp peer=ipadd remoteas=asn
    [authentication={md5|none}] [password=password]
    [description={none|description}]
    [ehops={default|1..255}] [fastfallover={no|yes}]
```

- To change an **individual** peer when the parameter **is** available in the template, first remove the template from the peer by using the command:

```
set bgp peer=ipadd policytemplate=
```

  Specifying **policytemplate=** like this with no number removes the template. The peer retains the template's settings. Then change the settings you need to for that peer by using the command:

```
set bgp peer=ipadd [other-options]
```

For information on resetting peers after modification, see "How to Modify BGP Peers (Without Templates)" .

How to Delete BGP Peers

Table 27-7: Procedure for deleting a BGP peer

Step	Command	Action
1	**disable bgp peer**={*ipadd*|all}	Disable the peer. Enabled peers cannot be deleted.
2	**delete bgp peer**={*ipadd*|all}	Delete the peer.

As soon as the peer session goes down, the router removes any routes it learned from that peer. Once the route selection timer expires, the router withdraws the routes from other peers to which it had advertised them.

Optimising BGP

This section describes a number of ways in which you can optimise BGP:

- How to Minimise the Impact of Unstable EBGP Routes
- How to Withdraw Routes As Soon As they Fail
- How to Advertise as Few Routes as Possible
- How to Improve IBGP Scalability
- How to Handle Spikes in Memory Use
- How to Stop BGP from Overloading System Memory
- How to Avoid Leaking Private AS Numbers into Global BGP Tables
- How to Set the IP Address that Identifies the Router

How to Minimise the Impact of Unstable EBGP Routes

Problem

BGP-managed networks are more efficient and stable when routing update messages are kept to a minimum. Under some network conditions, BGP generates an excessive rate of update messages due to "route flapping", in which some routes frequently oscillate between being reachable and unreachable. Route flapping causes a ripple effect through the BGP network as the changes are propagated. In extreme cases, the network does not reach a stable, converged state for a substantial period of time.

Solution: route flap damping

BGP route flap damping, as defined in RFC 2439, limits the impact and visibility of route flapping to a router's BGP peers. When a local BGP peer learns a route, it immediately adds it to its Routing Information Base (RIB) but may not immediately select or advertise it. It can only select or advertise the route once its internal BGP suppression engine considers that the route is sufficiently stable. A new route has no history of instability, so is immediately made available as normal. A route that has been previously learned but withdrawn, however, may be suppressed for a period of time, based on the severity of its previous instability. Therefore, BGP route flap damping suppresses routes that are considered too unstable to be used locally or advertised to any BGP peers, until the route has remained stable for a sufficient period of time. Persistently unstable routes may be excluded from selection indefinitely. By taking into account the prior behaviour of that route, the router can estimate the future stability of the route accurately enough to reduce router processing load without significantly impacting the convergence time for more stable routes.

Figure of Merit (FoM)

A route's history of instability is recorded via the maintenance of a statistic defined as a Figure of Merit (FoM) by RFC 2439. The FoM for a particular BGP route quantifies that route's history of stability, or lack of stability. When the router learns a new route, it grants the route an initial FoM of zero, indicating no history of instability. At this point, the BGP suppression engine is not interested in the route.

If an EBGP peer ever withdraws a route, the router increments the FoM for that route by 1000. As soon as a route earns a non-zero instability metric, the BGP suppression engine begins to monitor it, but in most configurations takes no other action at this stage. If the route exhibits further instability, its FoM

increases. Once the FoM exceeds a configurable suppression threshold, the BGP suppression engine begins to suppress it. At this stage, BGP no longer selects or advertises the route. Each route's FoM is reduced over time at a configurable rate, so if a suppressed route remains stable for a sufficient period of time, its status is eventually downgraded to monitored. If a monitored route's FoM reaches zero, or a value very close to zero, monitoring stops until further instability is observed. Figure 27-9 on page 27-25 shows the states and progress between them. Figure 27-10 on page 27-25 shows how a route's FoM is maintained over time.

Figure 27-9: The process applied to routes by route flap damping

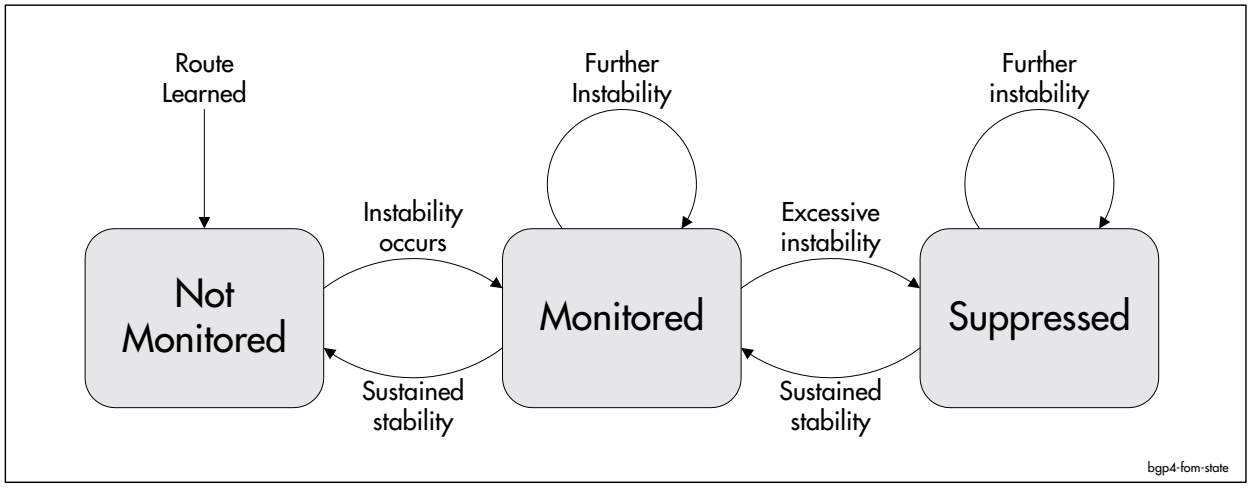
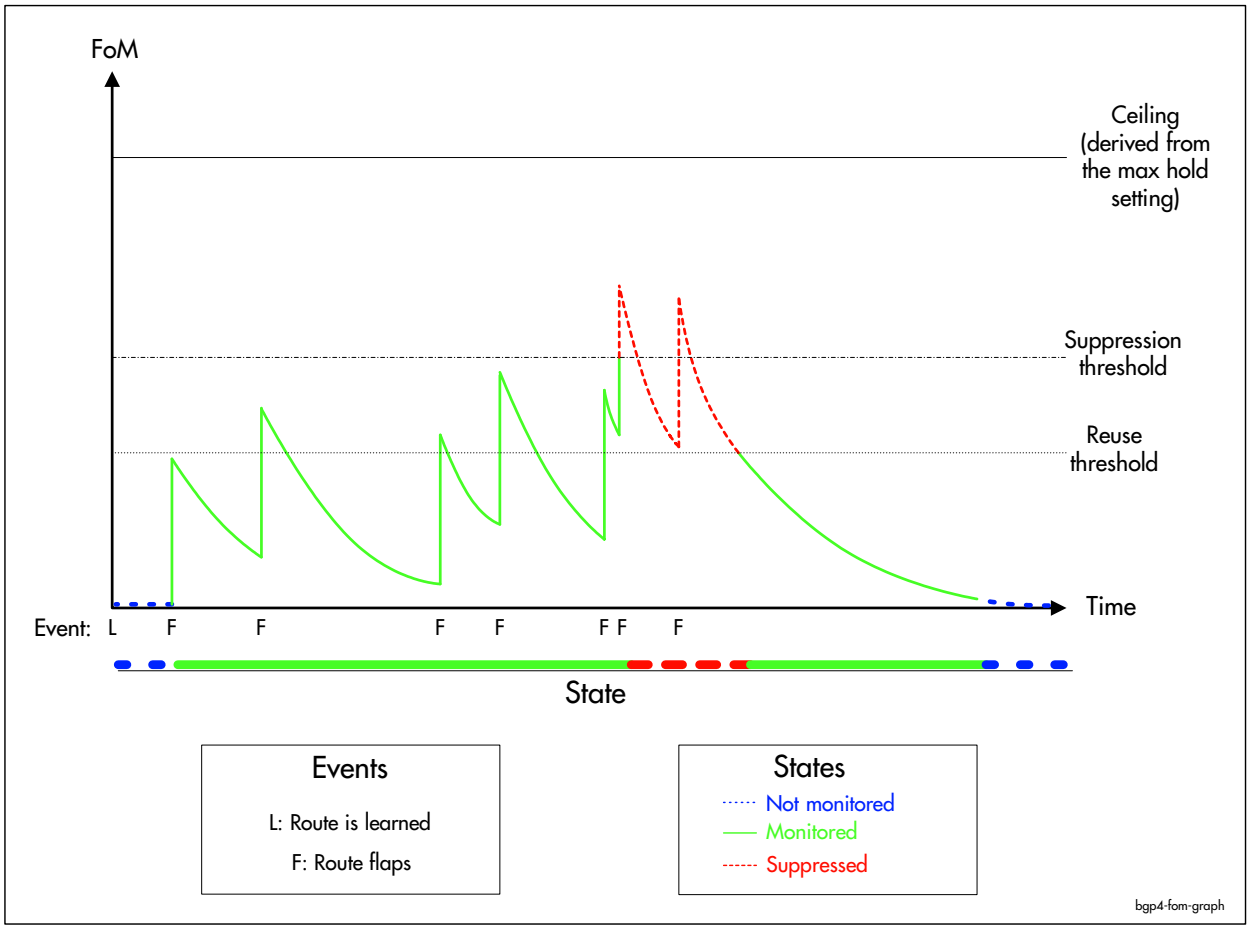


Figure 27-10: Change in FoM over time as a route flaps, showing when the route is suppressed



How to Configure Route Flap Damping

BGP route flap damping is disabled by default. You can enable it on all routes, or limit it to routes received by particular peers. You can use the default threshold settings, or specify different settings. The settings are captured by damping **parameter sets**, which are collections of four configuration parameters that determine the nature of the treatment received by relevant routes from the BGP suppression engine. Table 27-8 shows the parameters and the effect of increasing or decreasing each value.

Table 27-8: The effect of modifying route flap damping parameters

Parameter	Meaning of parameter	Change	Effect of change
suppression	**Suppression** is an FoM value. When a route's FoM exceeds this threshold, the route is suppressed.	Raised	Increases the number of times the route can become unreachable before it is suppressed.
		Lowered	Decreases the number of times the route can become unreachable before it is suppressed.
reuse	**Reuse** is an FoM value. Once a route is suppressed, it remains suppressed until its FoM falls below this threshold.	Raised	Decreases the minimum time that the route is suppressed.
		Lowered	Increases the minimum time that the route is suppressed.
halflife	**Halflife** is the time interval within which the route's FoM will halve if the route remains stable. For example, if the halflife is 15, the FoM of a stable route reduces by 50% over a 15 minute period, 75% over a 30 minute period, and so on.	Lengthened	Lowers the FoM more slowly, so increases the time the route is suppressed.
		Shortened	Lowers the FoM more quickly, so decreases the time the route is suppressed.
maxhold	**Maxhold** multiplied by **halflife** is the maximum period of time that a route must remain stable in order to become unsuppressed. For example, if **halflife** is 15 and **maxhold** is 4, the route is unsuppressed after 60 minutes of stability even if its FoM still exceeds **reuse**.	Increased	Increases the time that a severely unstable route must be stable, before it is unsuppressed.
		Reduced	Decreases the time that a severely unstable route must be stable, before it is unsuppressed.

Damping all routes

To apply route flap damping to all incoming routes by using the default parameter settings, use one of the commands:

```
enable bgp damping
enable bgp damping parameterset=0
```

If you do not want to use the default parameter settings, change the settings for the default parameter set 0 by using the command:

```
set bgp damping parameterset=0 [description=description]
    [suppression={default|1..20000}]
    [reuse={default|1..20000}] [halflife={default|1..45}]
    [maxhold={default|1..8}]
```

For more information about the default parameter set, see "Default parameter set" on page 27-27.

When you enable route flap damping globally, the router examines the instability history of every route it receives from every remote peer, and suppresses the route if appropriate.

Damping routes on specific peers

To limit route flap damping to some or all routes received by a particular peer, follow this procedure:

Table 27-9: Procedure for damping routes on specific peers

Step	Command	Action
1	**create bgp damping parameterset**=1..100 [description=*description*] [suppression={default|1..20000}] [reuse={default|1..20000}] [halflife={default|1..45}] [maxhold={default|1..8}]	Create a parameter set.
2	**enable bgp damping** parameterset=1..100	Enable BGP damping for the parameter set. You must enable only the desired parameter set, or route flap damping applies to all routes.
3	**show bgp damping**	Check the state and settings of the parameter set.
4	**add ip routemap**=*routemap* entry=1..4294967295 [action={include|exclude}] match aspath=1..99 **add ip routemap**=*routemap* entry=1..4294967295 [action={include|exclude}] match community=1..99 [exact={no|yes}] **add ip routemap**=*routemap* entry=1..4294967295 [action={include|exclude}] match nexthop=*ipadd* **add ip routemap**=*routemap* entry=1..4294967295 [action={include|exclude}] match origin={egp|igp|incomplete} **add ip routemap**=*routemap* entry=1..4294967295 [action={include|exclude}] match prefixlist=*name*	Create a route map to match the routes to which you want to apply route flap damping. If you are using an AS path list or community list to match routes, also configure the list. See "Creating Route Maps for BGP Routes" on page 28-10 of Chapter 28, Filtering IP Routes for more information.
5	**add ip routemap**=*routemap* entry=1..4294967295 set bgpdampid=1..100	Associate the damping parameter set with routes that match the route map. The **bgpdampid** parameter is the number of the route flap damping parameter set.
6	set bgp peer=*ipadd* inroutemap=*routemap*	Configure the BGP peer to use the route map on update messages it receives.

Default parameter set

If you enable route flap damping, all routes to which you do not specifically apply a parameter set are processed by the default parameter set. This set is numbered 0, and by default has the following settings:

- suppression=2000
- reuse=750
- halflife=15
- maxhold=4

The purpose of the default parameter set is to suppress routes that are not processed by any other parameter set, as shown in Figure 27-11 on page 27-28. Therefore you cannot limit the default parameter set to routes received by particular peers.

You can disable the default parameter set without disabling the whole of BGP damping by using the command:

```
disable bgp damping parameterset=0
```

You cannot destroy the default parameter set, but you can modify its settings by using the command:

```
set bgp damping parameterset=0 [description=description]
   [suppression={default|1..20000}]
   [reuse={default|1..20000}] [halflife={default|1..45}]
   [maxhold={default|1..8}]
```

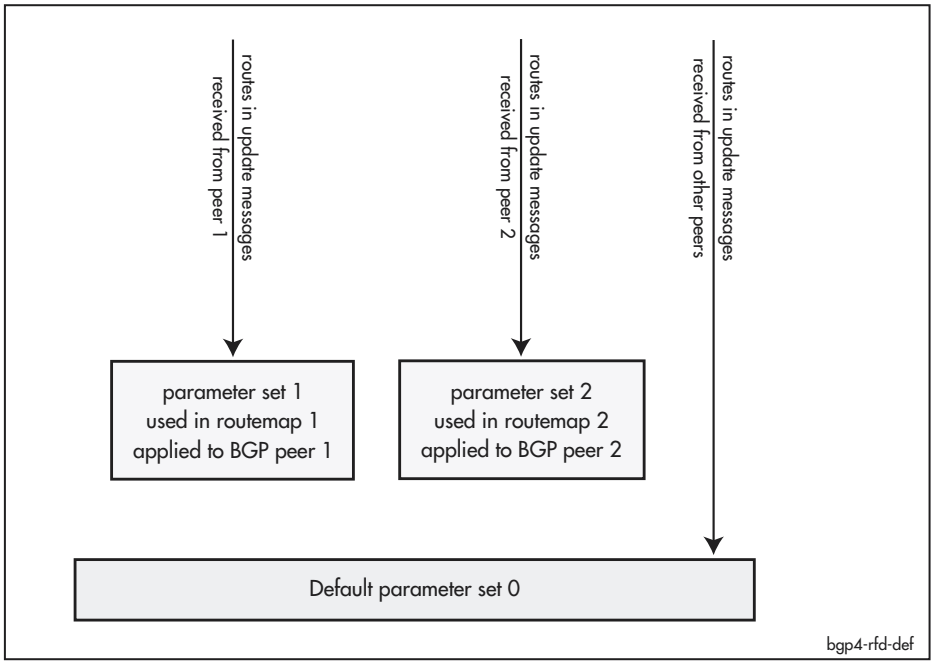
You can return the modified default parameter set to its original values by using the command:

```
set bgp damping parameterset=0 suppression=2000 reuse=750
   halflife=15 maxhold=4
```

Alternatively, you can reset the default parameter set and at the same time destroy all other parameter sets and clear all instability history. To do this, use the command:

```
purge bgp damping
```

Figure 27-11: Use of certain parameter sets on some peers and the default set on all others



Displaying route flap damping information

To display the current state of route flap damping and each parameter set, and the parameter values for each parameter set, use the command:

```
show bgp damping
```

To display a list of all monitored and suppressed routes, including their current FoM and the period of time which they must remain stable in order to progress from suppressed to monitored, or from monitored to not monitored, use the command:

```
show bgp damping routes
```

How to Withdraw Routes As Soon As they Fail

Problem

By default, when the interface that supports an EBGP peer session goes down, the corresponding peer session is not reset until that session's hold timer expires.

Solution: fast fallover

Fast failover is an option that you can enable for individual peers, that resets the session as soon as the router's interface to the peer goes down. It provides fast failover in case of link failures, because the router withdraws paths as soon as the link goes down, rather than waiting for up to three minutes to propagate the change. As a result, fast fallover greatly improves the rate of convergence to new network topology.

How to configure fast fallover

Fast fallover is disabled by default. To enable fast fallover on the router's link to a peer, use one of the commands:

```
add bgp peer=ipadd remoteas=1..65534 fastfallover=yes
    [other-options]
```

```
set bgp peer=ipadd fastfallover=yes [other-options]
```

To disable fast fallover on the router's link to a peer, use the command:

```
set bgp peer=ipadd fastfallover=no [other-options]
```

To see if fast fallover is enabled on the router's link to a peer, use the command:

```
show bgp peer=ipadd
```

What about unstable links?

Certain types of links can be particularly unreliable. If you enable fast fallover on such a link, the BGP sessions supported by the link will flap. This causes frequent route changes and excessive update messaging within the network.

If a link to a peer is susceptible to brief outages, we do not recommend enabling fast fallover on it. If it is susceptible to longer outages, fast fallover may be desirable because it lets the router rapidly change to an alternative path.

Using it with VLANs

BGP peer fast fallover recognises that a link has gone down when the relevant layer 2 interface notifies BGP that its link status has changed from up to down. A VLAN only changes its link status to down when all router ports in the VLAN are down. If the router connects to a peer through a port in a VLAN, and the VLAN also contains other ports, the link to the peer may go down without changing the VLAN status, so BGP cannot tell that the link is down. This happens when the port connected to the peer goes down but unrelated ports in the VLAN are still up. Therefore, we recommend you only apply fast fallover to a peer reached through a VLAN when all ports in the VLAN connect to the peer.

How to Advertise as Few Routes as Possible

Problem

When BGP learns routes, it imports and advertises every route, even if some are routes to subnets of the same network. For example, if you used the subnets 192.168.1.64/26 and 192.168.1.128/26, BGP would advertise routes to both of these.

Depending on the router's role in your network, this may be undesirable because it:

- exposes network topology
- creates more update messages than necessary
- increases the size of the routing table

Solution

There are two available solutions:

- Route aggregation, which is useful when you want to summarise subnet routes that are within particular class A, B or C networks. This option allows BGP to summarise subnets from any source, including from BGP peers.
- Automatic summarising, which enables BGP to automatically summarise all locally-originated prefixes into their class A, B or C networks. This option allows BGP to summarise prefixes when it imports OSPF, RIP, interface and statically-configured routes.

Aggregating Routes

About route aggregation

When BGP receives routes from its peers or imports them from the RIB, by default it advertises every route, no matter how specific. You can reduce the number of routes BGP advertises, by configuring aggregate prefix entries. If the router receives a route to a subset of the entry's prefix, BGP adds the aggregate prefix to its database, as well as the route for the more specific prefix. You can set the router to advertise only the aggregate.

Consider a configuration in which you create an aggregate entry of 192.168.1.0/24 and set the aggregate entry to advertise only the aggregate. If the router receives routes to the prefixes 192.168.1.64/26 and 192.168.1.128/26, it stores all three prefixes but only advertises 192.168.1.0/24.

Note that the router does not use the aggregate route for IP routing. The router only uses the aggregate to determine which routes to advertise.

The router advertises the aggregate route as coming from the router's autonomous system, and sets the aggregate's atomic_aggregate attribute.

Caution Make sure that you own all the IP addresses in the aggregate entry. Otherwise, you advertise yourself as the next hop to addresses that you do not own.
For example, if you own 202.202.202.0/24, you can configure that as an aggregate entry. However, if you only own 202.202.202.64/26, you must not configure an aggregate of 202.202.202.0/24.

Configuring route aggregation

To aggregate subnets and only advertise the aggregate prefix, use the command:

```
add bgp aggregate=prefix[/0..32] [mask=mask] summary=yes
    [routemap=routemap]
```

The **aggregate** parameter specifies the network into which BGP aggregates subnets.

The **summary** parameter controls advertisement. If **summary** is **yes**, the router only advertises the route to the aggregate. Note that unadvertised routes are still displayed in output of the **show bgp route** command, but are marked with an “s”.

Creating an aggregate entry does not immediately add the aggregate prefix to the BGP routing table. BGP adds the aggregate prefix when it receives an advertisement of a more specific subnet.

Automatic Summarising

About automatic summarising

When BGP imports routes from another routing source, such as OSPF, by default it stores and advertises every route, no matter how specific. If your LAN is divided into subnets, this means BGP advertises a route to each subnet. You can avoid this by enabling automatic summarising. This feature summarises prefixes into networks and only advertises a route to that network. It is particularly useful on the external speaker for an AS—the router that links an internal network to a public network.

When you enable automatic summarising, the router summarises subnets into their Class A, B or C network. Instead of writing the route to the subnet into the BGP routing table and advertising that subnet route, it writes a single route to the summary network. For example, instead of storing and advertising routes to 192.168.1.64/26 and 192.168.1.128/26, BGP would have one route to 192.168.1.0/24.

Caution Only turn on automatic summarising if you own the whole classful network for your locally-generated routes. Otherwise, you advertise yourself as the next hop for subnets that you do not own.
For example, if you owned 202.202.202.0/24, you could use automatic summarising. However, if you only owned 202.202.202.64/26, you must not use automatic summarising.

Configuring automatic summarising

If you want to import routes from RIB into BGP and automatically summarise them into networks, use the following procedure. Instead of importing routes to subnets within each network, BGP then imports and advertises the route to the summary network. It specifies this router as the next hop for the summary route.

Step	Command	Action
1	**add bgp** **import**={interface|ospf|rip|static} [routemap=*routemap*] or **add bgp network**=*prefix*[/0..32] [mask=*mask*] [routemap=*routemap*]	Turn on importing for the required routing source or network. Note that automatic summarising applies to all routes that BGP imports. If you configure multiple import or network entries, BGP summarises routes from all of them.
2	**enable bgp autosummary**	Enable automatic summarising.
3	**show bgp route**	Check that BGP has imported and summarised the desired networks.

Examples of automatic summarising

The following table uses the example of the static routes 192.168.1.64/26 and 192.168.1.128/26 to show what BGP advertises with different combinations of import and network entries, with and without automatic summarising.

Automatic summarising?	Commands	BGP advertises
No	add bgp import=static add ip route=192.168.1.64/26 nexthop=*ipadd* add ip route=192.168.1.128/26 nexthop=*ipadd*	Routes to 192.168.1.64/26 and 192.168.1.128/26.
	add bgp network=192.168.1.0/24 add ip route=192.168.1.64/26 nexthop=*ipadd* add ip route=192.168.1.128/26 nexthop=*ipadd*	Nothing. BGP does not advertise a route to 192.168.1.0/24 unless it can find one in the router's RIB.
	add bgp import=static add bgp network=192.168.1.0/24 add ip route=192.168.1.64/26 nexthop=*ipadd* add ip route=192.168.1.128/26 nexthop=*ipadd*	Routes to 192.168.1.64/26 and 192.168.1.128/26. BGP does not advertise a route to 192.168.1.0/24 unless it can find one in the router's RIB.
Yes	add bgp import=static enable bgp autosummary add ip route=192.168.1.64/26 nexthop=*ipadd* add ip route=192.168.1.128/26 nexthop=*ipadd*	A single route to 192.168.1.0/24 with nexthop=0.0.0.0. Automatic summarising replaces the two subnet entries in the BGP routing table with this one entry.
	add bgp network=192.168.1.0/24 enable bgp autosummary add ip route=192.168.1.64/26 nexthop=*ipadd* add ip route=192.168.1.128/26 nexthop=*ipadd*	A single route to 192.168.1.0/24 with nexthop=0.0.0.0. BGP advertises 192.168.1.0/24 because it finds a route to that network in the router's RIB.
	add bgp import=static add bgp network=192.168.1.0/24 enable bgp autosummary add ip route=192.168.1.64/26 nexthop=*ipadd* add ip route=192.168.1.128/26 nexthop=*ipadd*	A single route to 192.168.1.0/24 with nexthop=0.0.0.0. You do not need to specify both import and network entries.

How to Improve IBGP Scalability

Problem

If a BGP peer learns a route from an EBGP peer and selects it as the best available route to the given destination network, it sends an update message advertising the route to all its IBGP and EBGP peers. However, if a peer learns a route from an IBGP peer, it does not send an advertisement to its other IBGP peers. This policy requires that all BGP speakers within an autonomous system be fully meshed—each internal speaker must be connected to every other internal speaker. As a result, the scalability of a BGP autonomous system is in the order of n^2 (n speakers require $n(n\text{-}1)/2$ peer sessions).

Solution: route reflection

BGP Route Reflection improves the scalability of the AS, by giving specific IBGP peers the authority to advertise IBGP-learned routes to a predefined subset of their IBGP peers. Route Reflection is defined in RFC 2796, *BGP Route Reflection—An Alternative to Full Mesh IBGP*. As shown in Figure 27-12, an AS using route reflection consists of at least one router that advertises IBGP-learned routes, called a *Route Reflector* (RR), and that router's IBGP peers. Each peer is one of the following types:

- *Client Peer* (CP)

  Client peers maintain IBGP peer sessions only with one or more of the RRs of their AS. CPs rely on an RR to advertise routes that they originate to the other members of the AS. When an RR receives a route from a client peer, the RR reflects the route to all its peers, both client and non-client.

- *Non-Client Peer* (NCP)

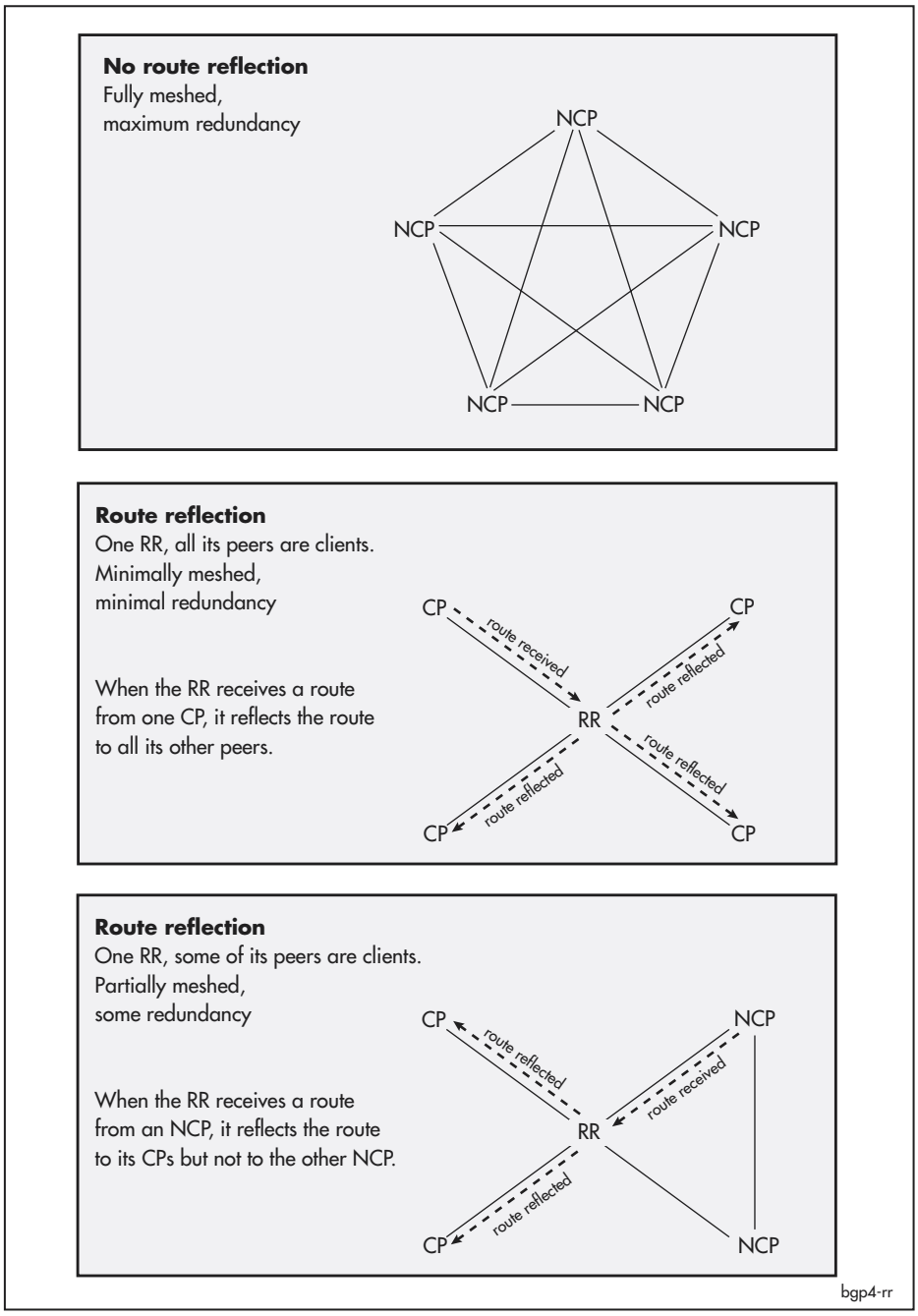
  Non-client peers maintain peer sessions with both the RRs of their AS, and all other non-client peers in the AS. NCPs only rely on an RR to advertise routes to the RR's client peers. When an RR receives a route from a non-client peer, the RR reflects the route only to its client peers, not to other non-client peers it has.

  This means that client peers are not required to be connected to each other but non-client peers are.

Note that route reflection does not affect the route selection process; it simply determines the peers to which the selected routes are advertised.

Tip You can also use BGP confederations instead of full-meshing or route reflection. However, route reflection has the advantage that only the BGP hosts that perform the reflection need to understand route reflection. All hosts in a confederation need to be confederation-aware.

Figure 27-12: An IBGP AS of 5 routers, with and without route reflection



How to configure route reflection

To configure route reflection, follow this procedure:

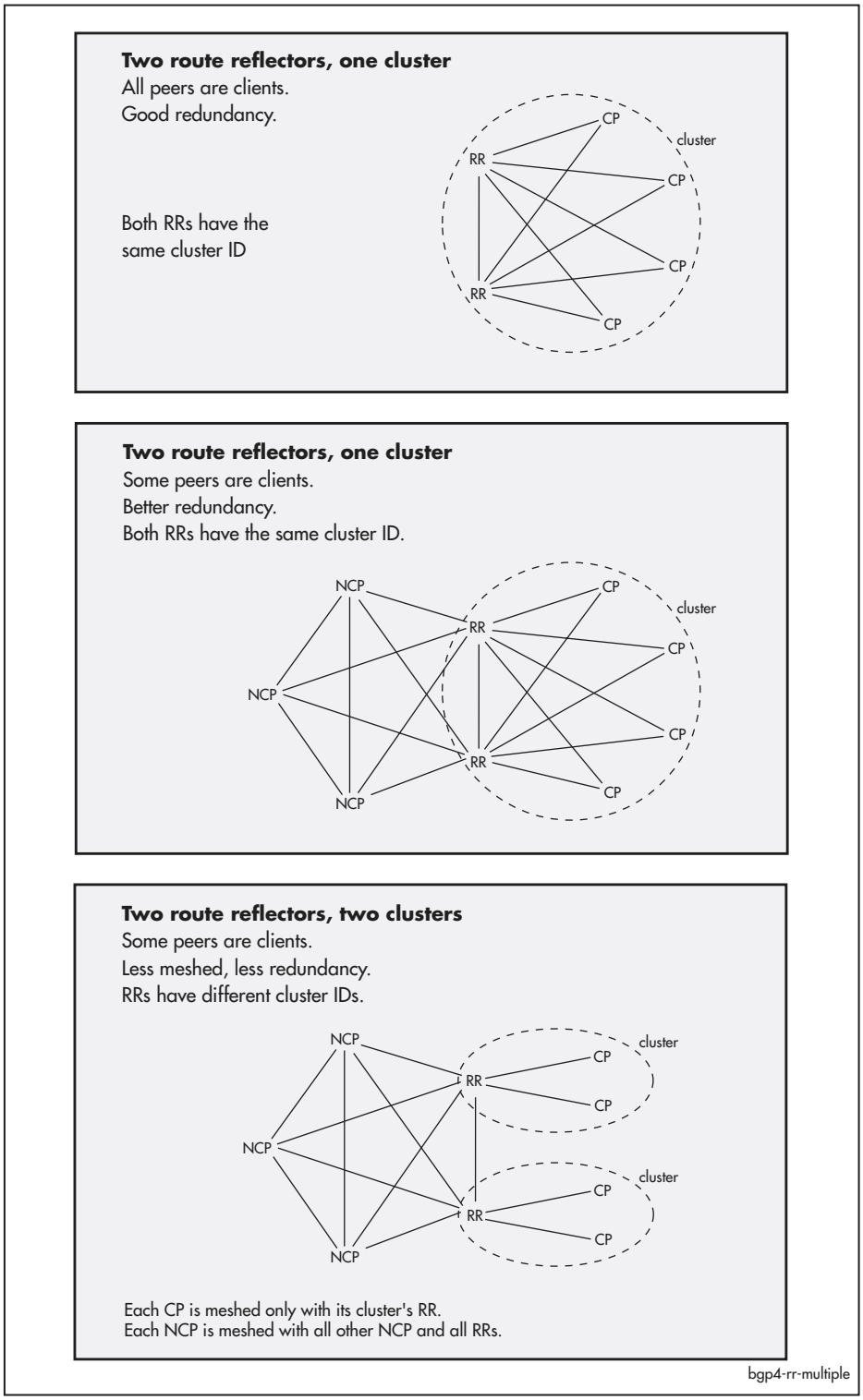
Table 27-10: Procedure for configuring route reflection

Step	Command	Action
1	—	Determine which router in your AS will be the RR and which routers will be client peers of the RR.
2	—	Connect the routers in your IBGP AS together so that each peer is directly connected to the RR. Non-client peers must also be connected to each other—any non-client part of the AS must be fully meshed because NCPs do not re-advertise routes among themselves.
3	**add bgp peer**=*ipadd* remoteas=*asn* client={no|yes} [*other-options*] or **set bgp peer**=*ipadd* client={no|yes} [*other-options*]	On the router that is the RR, configure peer relationships to each of the other routers, specifying whether each peer is a client or non-client. The default for the **client** parameter is **no**, making the peer a non-client peer. The router is an RR if it has at least one client peer.
4	**add bgp peer**=*ipadd* remoteas=*asn* [client=no] [*other-options*] or **set bgp peer**=*ipadd* [client=no] [*other-options*]	On the other routers, configure peer relationships to the route reflector, and to any other peers. Note that the RR is a non-client peer of its clients.

RR redundancy

Although a full-mesh AS suffers from poor scalability, it has the advantage of being extremely robust. In contrast, an AS that employs a single RR to serve a large number of clients is relatively vulnerable to congestion or loss of connectivity, because the RR plays a critical role in the operation of the AS. As shown in Figure 27-12 on page 27-34, you can reduce the vulnerability by making parts of the AS fully meshed. Alternatively, you can have multiple RRs within an AS by configuring one or more *clusters* (Figure 27-13 on page 27-36).

Figure 27-13: Multiple RRs in an AS



When there are multiple RRs in an AS, routing information loops become possible. The following table describes two BGP attributes that route reflection uses to detect and prevent loops.

Attribute	Length	Type code	Definition and use	User-set
Originator_ID	4-byte	9	An attribute that identifies the first local AS member to learn the route. When the router is an RR and receives a new route from an IBGP peer, it adds the originator_ID attribute if one was not already present. The router uses the router ID of the IBGP peer that received the given route from an EBGP peer. If a CP or NCP receives an update that contains its own router ID as the originator ID, it ignores the update.	No
Cluster_list	variable	10	A list of 4-byte cluster_ID values that together represent the reflection path of the given route through the local AS. When an RR reflects a route, it adds its cluster_ID to the cluster_list. If a CP or NCP receives an update that contains its router ID as the originator, it ignores the update instead of reflecting it. This prevents routing information loops.	Cluster_ID is configurable.

CPs and NCPs forward these attributes unchanged within their AS. The router removes the attributes from updates that are destined for EBGP peers.

How to configure multiple RRs

Use the following procedure to configure multiple RRs in an AS.

Table 27-11: Procedure for configuring multiple RRs

Step	Command	Action
1	—	Partition the AS into clusters. Each cluster has at least one RR and at least one CP.
2	**set bgp** cluster=*ipadd* [*other-options*]	Give all RRs in each cluster the same CLUSTER_ID so they can avoid routing information loops. Use the router ID of one of the RRs as the CLUSTER_ID.
3	**add bgp peer**=*ipadd* remoteas=*asn* client={no|yes} [*other-options*] or **set bgp peer**=*ipadd* client={no|yes} [*other-options*]	Configure the required peer relationships on all routers in the AS (see "How to configure route reflection" on page 27-35). Configure the RRs as peers of each other so each RR can reflect the other RRs' routes.

How to Handle Spikes in Memory Use

Problem

While BGP is running, other software modules may cause a spike, or surge, of system memory utilisation for brief periods of time.

Solution: BGP backoff

BGP backoff elegantly handles low system memory situations. When memory is heavily used, BGP delays its processing until system memory is more abundant.

The backoff utility allows other processes access to the memory resources they need, without actually shutting BGP down unless it determines that BGP has backed off for a prolonged period of time. By default, BGP delays its processing for 10 seconds if system memory utilisation reaches 95%, and stays backed off until memory usage drops to 90%.

How to configure BGP backoff

To change BGP system memory backoff settings, use the command:

```
set bgp backoff[=20..100] [basetime=0..100]
   [consecutive=0..1000] [low=15..99] [multiplier=1..1000]
   [step=1..1000] [totallimit=0..1000]
```

This command provides the following configuration options:

- thresholds of total system memory utilisation that causes BGP to trigger and maintain back off—the **backoff** and **low** parameters (see "Thresholds" below)
- time that BGP backs off for—a combination of the **step**, **basetime** and **multiplier** parameters (see "Backoff time" below)
- total number of backoffs before all BGP peers are disabled—the **totallimit** parameter
- total consecutive number of backoffs before all BGP peers are disabled—the **consecutive** parameter.

Thresholds

BGP backoff has upper and lower thresholds for the percentage of system memory use which triggers and maintains BGP backoff. When memory usage exceeds the upper threshold, BGP backoff is triggered. BGP continues to back off until memory usage falls below the lower threshold. At this stage BGP begins processing again, unless the total or consecutive backoff limits were reached.

The upper threshold is set using the **backoff** parameter, and must be a higher percentage than the lower threshold. The lower threshold is set using the **low** parameter, and must be a lower percentage than the upper threshold. The **backoff** and **low** parameters cannot be set at the same value.

As the router will not allow the **backoff** parameter value to be set below the **low** parameter, we recommend that you always adjust these parameters in the same command. For example:

```
set bgp backoff=88 low=84
```

Backoff time

The backoff time is recalculated after a given number of backoffs—this is called a *step*. The first backoff time is calculated as:

```
base time x multiplier/100
```

Backoff time is recalculated after each step, based on the current backoff time:

```
current backoff time x multiplier/100
```

The value is rounded down to the nearest second (unless it is less than 1 second, in which case it is set to 1 second).

For example, a base time of 60 seconds with a multiplier of 110 increases the timeout by 10 percent every time the backoff time is recalculated. Thus, a step value of 2 and multiplier of 110 results in the numbers in the following table.

Backoff Iteration	Time to Backoff (secs)
0	60
1	60
2	66
3	66
4	72
5	72
6	79
7	79

A multiplier of less than 100 percent gives the effect of a decay mechanism, and a multiplier of greater than 100 percent gives the effect of an accumulative mechanism.

Consecutive backoffs

If BGP gets to the end of the backoff period and system memory is still above the lower memory use threshold, BGP backs off immediately without performing any processing. Such backoffs are called *consecutive backoffs*.

The router counts the number of consecutive backoffs. It resets the count to zero whenever BGP is able to perform some processing after a backoff. By default, the number of consecutive backoffs is limited to 5. After BGP reaches this limit, the router considers that BGP is irrecoverable and disables all peers. You then need to manually reinstate the peers, by using the **enable bgp peer** command on page 27-81 command.

Enabling or disabling BGP backoff

BGP backoff can be enabled or disabled using the commands **enable bgp backoff** command on page 27-77 and **disable bgp backoff** command on page 27-72. BGP backoff is disabled by default, however it automatically enables the first time a peer is added.

How to Stop BGP from Overloading System Memory

BGP memory accounting limits how much of the total system memory can be in use by BGP and IP routing combined. The router disables BGP if it uses more than the set percentage of system memory it has been allocated. This also shuts down BGP peers, and therefore all routes learnt from those peers are dropped. The default amount of system memory allocated to BGP and IP routing is 85%.

To change the memory limit, use the command:

```
set bgp memlimit[=0..100]
```

To see the current limit and usage, use the command:

```
show bgp memlimit
```

To see detailed technical information about memory usage, use the command:

```
show bgp memlimit scan
```

How to Avoid Leaking Private AS Numbers into Global BGP Tables

AS numbers are two bytes in length, so range from 1 to 65535. Of this value range, the AS numbers 1 to 64511 are globally unique and are assigned by InterNIC. The remaining value range from 64512 to 65535 is reserved for AS numbers that are private. These numbers are unique only within the scope of a given administrative domain.

Because private AS numbers are not globally unique, they should not be leaked to global BGP routing tables, in which context they become ambiguous. To prevent private AS numbers from crossing administrative boundaries, the router supports the stripping of private AS numbers from the AS Path attributes of outgoing update messages. You can configure this on a per-peer or per-template basis. It is disabled by default. To configure a peer, use one of the commands:

```
add bgp peer=ipadd  remoteas=asn privateasfilter={yes|no}
    [other-options]

set bgp peer=ipadd  privateasfilter={yes|no} [other-options]
```

To configure a peer template, use one of the commands:

```
add bgp peertemplate=1..30 privateasfilter={yes|no}
    [other-options]

set bgp peertemplate=1..30 privateasfilter={yes|no}
    [other-options]
```

How to Set the IP Address that Identifies the Router

When the router is acting as a BGP speaker, it uses an IP address to identify itself to its peers in these situations:

- when establishing the TCP session and sending TCP messages
- in the *open* message it sends at the beginning of the session
- when it considers itself to be the next hop for a route that it is advertising to its peers.

Address selection rules

The address the router uses in each of these situations depends on the situation and whether you have configured a router ID or a local interface address. The rules for each situation are:

1. TCP session source address

   If a local IP address has been set for the peer, use it. Otherwise allow TCP to select a source IP address, which it does based on the outgoing interface.

2. BGP Identifier in *open* message

   If the router ID has been set, use it. Otherwise, if a local IP address has been set for the peer, use that. If neither has been set, use the highest IP address configured on any of the router's interfaces.

3. Next hop address

   If the router learned the route from an IBGP peer, use the learned next hop address—the next hop that the IBGP peer supplied for the route.

If the router learned the route from an EBGP peer and the learned next hop is in the same subnet as the router, use the learned next hop.

If the router learned the route from an EBGP peer and the learned next hop is in a different subnet to the router, then:

- if a local IP address has been set for the peer to which the router is sending the update, use it
- otherwise, if the router has an IP route to that network, use the IP address of the interface via which the route reaches that network
- otherwise, use the IP address of the interface via which the router reaches the peer to which it is sending the update

How to configure router ID

To configure a router ID, use the command:

```
set bgp routerid=ipadd [other-options...]
```

How to configure local interface

To configure a local interface, first create the local interface and give it an IP address by using the command:

```
add ip local=1..15 ipaddress=ipadd [other-options...]
```

Then apply the local interface to the BGP peer by using one of the commands:

```
add bgp peer=ipadd remoteas=1..65534 local=1..15
   [other-options...]
```

```
set bgp peer=ipadd local=1..15 [other-options...]
```

Configuration Examples

This section includes the following configuration examples:

- **Basic BGP Configuration**
- **Advanced BGP Configuration**

Basic BGP Configuration

This example establishes a BGP session between two routers, A and B. Each router is an external BGP peer of the other, so each router is an external speaker.

Router A has an IP address of 10.0.0.2 and AS number of 2. Router B has an IP address of 10.0.0.1 and AS number of 1. The example assumes that the routers already have these IP addresses and can ping each other.

Important: This example uses private addresses in the 10.0.0.0 subnet instead of globally-unique IP addresses. Replace these addresses with suitable global addresses for your network.

The configuration is shown in:

- Figure 27-14—a diagram of the scenario
- Figure 27-15 on page 27-43—the commands to configure Router A
- Figure 27-16 on page 27-43—the commands to configure Router B
- "External speakers" on page 27-17—the general procedure for configuring external speakers

Figure 27-14: Example of a basic BGP-4 configuration

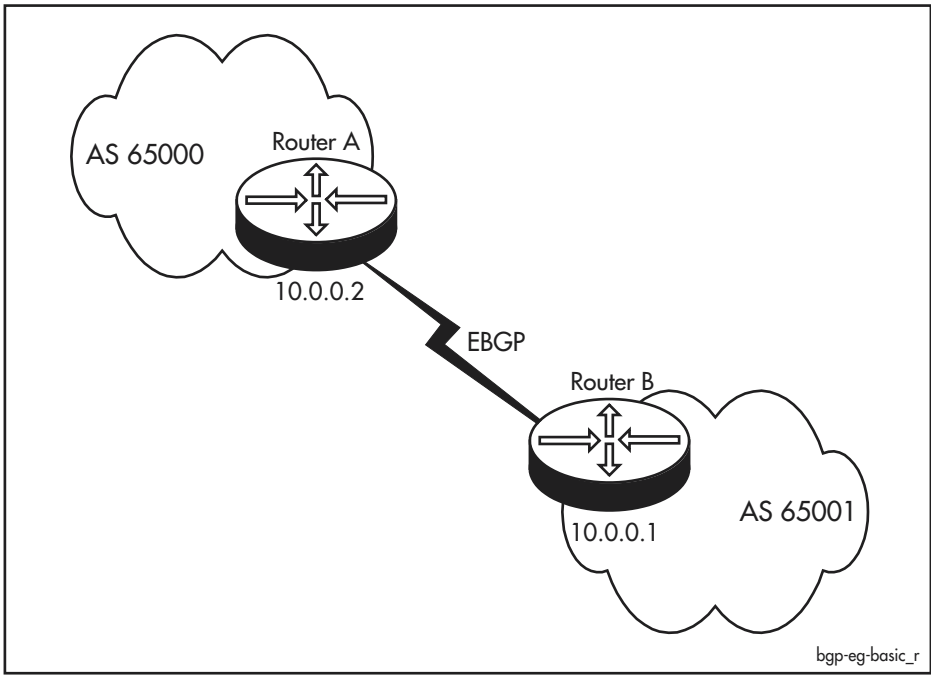


Figure 27-15: Example script for configuring Router A as an external BGP speaker

```
# Configuring Router A as an external BGP speaker
# BGP configuration only

# Set the AS number
set ip autonomous=65000

# Set the router ID
set bgp routerid=10.0.0.2

# Add Router B to Router A as a peer
add bgp peer=10.0.0.1 remoteas=65001

# Import interface routes
add bgp import=interface

# Enable the peer
enable bgp peer=10.0.0.1
```

Figure 27-16: Example script for configuring Router B as an external BGP speaker

```
# Configuring Router B as an external BGP speaker
# BGP configuration only

# Set the AS number
set ip autonomous=65001

# Set the router ID
set bgp routerid=10.0.0.1

# Add Router A to Router B as a peer
add bgp peer=10.0.0.2 remoteas=65000

# Enable the peer, to begin the BGP session
enable bgp peer=10.0.0.2

# Import interface routes
add bgp import=interface

# Verify the connection
show bgp peer
# Check that the peer 10.0.0.2 is present and has a State of Estab.
```

Advanced BGP Configuration

This example configures the router as an external BGP speaker and sets up a connection to a peer.

The example assumes that the router and its peer already have IP addresses and can ping each other.

Important: This example uses private addresses (in the 172.16.0.0 and 192.168.0.0 subnets) instead of globally-unique IP addresses. Replace these addresses with suitable global addresses for your network.

The configuration is shown in:

- Figure 27-17—a diagram of the scenario
- Figure 27-18 on page 27-45—the commands to configure Router A
- Figure 27-19 on page 27-46—the commands to configure Router B

The configuration includes these advanced features:

- setting up a peer relationship between routers that are in different networks. To do this, the configuration uses the **ehops** parameter of the **add bgp peer** command to specify the number of hops that separate the peers.
- basing the peer on a template (see "How to Create BGP Peers Using Peer Templates" on page 27-20). All peer definitions that use the template:
  - accept a maximum of 5000 prefixes
  - use the local IP interface local1 as the source IP address (see "How to Set the IP Address that Identifies the Router" on page 27-40)
- authenticating BGP messages
- using fast fallover (see "How to Withdraw Routes As Soon As they Fail" on page 27-29)
- using automatic summarising (see "How to Advertise as Few Routes as Possible" on page 27-30)

Figure 27-17: Example for a more advanced BGP-4 configuration

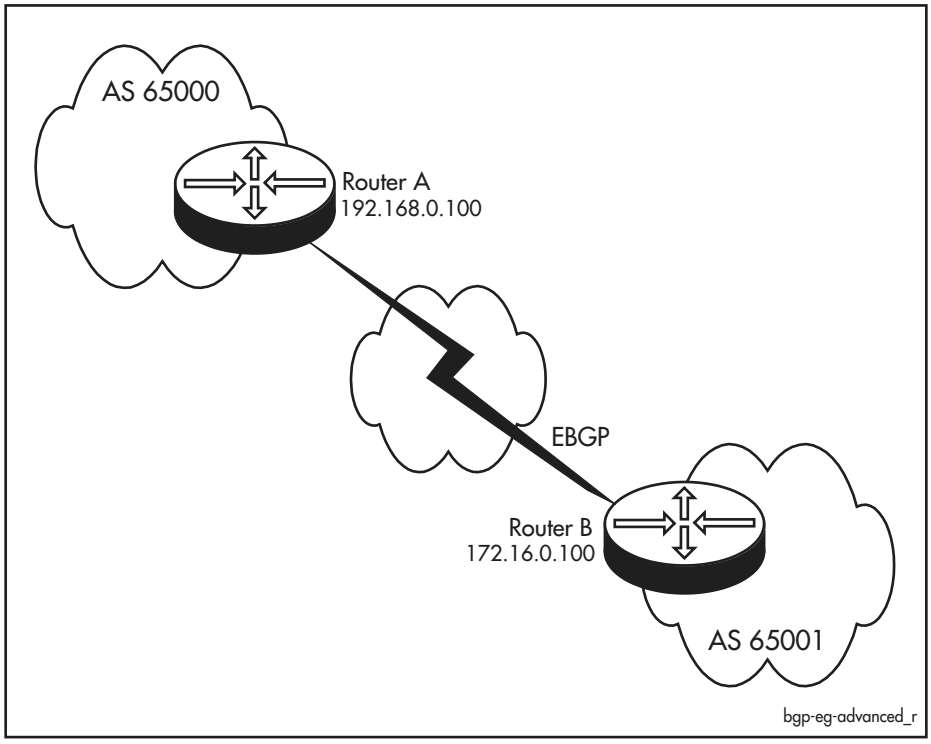


Figure 27-18: Example script for more advanced BGP configuration on Router A

```
# More advanced BGP configuration
# BGP and local IP interface configuration only

# Create a local interface
add ip local=1 ipaddress=192.168.0.100

# Create the peer template
add bgp peertemplate=1 local=1 maxprefix=5000 maxprefixaction=terminate

# Set the AS number
set ip autonomous=65000

# Create the peer, using the template
add bgp peer=172.16.0.100 remoteas=65001 policytemplate=1 fastfallover=yes ehops=3
authentication=md5 password=1verysecret

# Configure automatic summarising
enable bgp autosummary

# Import interface routes
add bgp import=interface

# Import statically-configured routes
add bgp import=static

# Enable the peer
enable bgp peer=172.16.0.100

# Verify the peer
show bgp peer
# Check that the peer 172.16.0.100 is present.
```

Figure 27-19: Example script for more advanced BGP configuration on Router B

```
# More advanced BGP configuration
# BGP and local IP interface configuration only

# Create a local interface
add ip local=1 ipaddress=172.16.0.100

# Create the peer template
add bgp peertemplate=1 local=1 maxprefix=5000 maxprefixaction=terminate

# Set the AS number
set ip autonomous=65001

# Create the peer, using the template
add bgp peer=192.168.0.100 remoteas=65000 policytemplate=1 fastfallover=yes ehops=3
authentication=md5 password=1verysecret

# Configure automatic summarising
enable bgp autosummary

# Import interface routes
add bgp import=interface

# Import statically-configured routes
add bgp import=static

# Enable the peer, to begin the BGP session
enable bgp peer=192.168.0.100

# Verify the connection
show bgp peer
# Check that the peer 192.168.0.100 is present and has a State of Estab.
show bgp route
# Check that expected static and interface routes are present.
# Check that BGP has received routes from the peer.
```

Command Reference

This section describes the commands available on the router to enable, configure, control and monitor BGP. See Chapter 22, Internet Protocol (IP) for the commands required to enable and configure IP to use BGP.

The shortest valid command is denoted by capital letters in the Syntax section. See "Conventions" on page lxv of About this Software Reference in the front of this manual for details of the conventions used to describe command syntax. See Appendix A, Messages for a complete list of messages and their meanings.

add bgp aggregate

Syntax

```
ADD BGP AGGregate=prefix[/0..32] [MASK=mask]
   [SUMmary={NO|YES}] [ROUTEMap=routemap]
```

Description

This command adds an aggregate entry to BGP. When a peer advertises a route that is a subset of the entry's prefix, the router adds the aggregate entry to its database, as well as the entry for the more specific route. This can increase the efficiency of BGP by allowing the router to process and advertise a single route instead of a large number of more specific subnets.

Note that the router does not use the aggregate route for IP routing. The router only uses the aggregate to determine which routes to advertise.

The router does not add the aggregate entry to its database until it receives an advertisement of a more specific subnet.

The router advertises the aggregate route as coming from the router's autonomous system, and sets the aggregate's atomic_aggregate attribute.

Parameter	Description
AGGregate	The network prefix to be used for this aggregate entry. This is expressed as the base IP address of the network, in dotted decimal notation, optionally followed by a "/" character and the number of bits in the network mask. If you do not specify the CIDR mask, the router uses the value from the **mask** parameter, if present, or otherwise the natural mask for the network, based on whether it is a class A, B, or C network. Default: no default
MASK	The network mask for the aggregate entry. This parameter is provided for compatibility with other router commands that specify an IP address and mask; we recommend that you instead specify the mask in the **aggregate** parameter. If you specify a mask in this parameter and the **aggregate** parameter, an error results unless the two masks agree. Default: The natural mask for the network, based on whether it is a class A, B, or C network

Parameter (cont)	**Description** (cont)	
SUMmary	Whether the router advertises only the aggregate route, or also the more specific routes that make up the aggregate. Default: **no**	
	NO	The router advertises the more specific routes that make up the aggregate.
	YES	The router only advertises the aggregate route. Note that unadvertised routes are still displayed in output of the **show bgp route** command, but are marked with an "s".
ROUTEMap	The route map used to filter the more specific routes that make up the aggregate, or to set attributes for the aggregate route. The *routemap* is the name of the appropriate pre-existing map. Default: no route map (routes are not filtered and attributes are not set)	

Examples

To add an aggregate entry for the network 198.168.0.0, with a mask of 255.255.0.0, use the command:

```
add bgp agg=192.168.0.0/16
```

As soon as the router learns a more specific route, such as 192.168.1.0/24, BGP also adds an entry for 192.168.0.0/16 to the routing table.

To add an aggregate entry for the network 192.168.8.0/21 and use route map *agg_map*, use the command:

```
add bgp agg=192.168.8.0/21 routem=agg_map
```

Related Commands

delete bgp aggregate
set bgp aggregate
show bgp aggregate
show bgp route

add bgp confederationpeer

Syntax

```
ADD BGP CONFEDerationpeer=1..65534
```

Description

This command adds an Autonomous System to the AS confederation to which this router belongs. An AS confederation is a group of Autonomous Systems which communicate between themselves using confederation BGP, and communicate to Autonomous Systems outside the confederation as if they were a single Autonomous System. For more information about AS confederations, see "AS Confederations" .

The **confederationpeer** parameter specifies the number of an Autonomous System that is to be treated as one of the Autonomous Systems in the confederation. This number cannot be the same as this router's AS number, or this router's confederation ID. The specified AS number should not already have been added with this command.

A router need not be configured with all of the AS numbers in the AS confederation, but only those with which it is to have peer relationships. Similarly, the confederation ID for the router only has to be configured on routers that are to have peer relationships with BGP routers outside the confederation.

When you create the peer relationship by using the **add bgp peer** command on page 27-52, specify the peer's confederation ID in the **remoteas** parameter.

Examples

To set up a confederation with AS numbers 65502, 65503 and 65504, whose external AS number is 1234, and with this router in AS 65501, use the commands:

```
set ip au=1234
set bgp conf=65501
add bgp confed=65502
add bgp confed=65503
add bgp confed=65504
```

Related Commands

add bgp peer
delete bgp confederationpeer
set bgp
set ip autonomous
show bgp confederation

add bgp import

Syntax

```
ADD BGP IMPort={INTerface|OSPF|RIP|STAtic}
    [ROUTEMap=routemap]
```

Description

This command adds an import entry to BGP. This instructs BGP to import routes from a given route source into the BGP route table. Optionally, you can specify a route map to allow filtering of routes and setting of BGP attributes.

When BGP imports routes from a protocol, it only imports routes that are the best routes to their destination networks. BGP determines that a route is the best route if:

- the route goes over an active interface
- that interface does not have an infinite or unreachable route
- the route has a higher routing preference metric than other candidate routes.

Parameter	Description	
IMPort	The source of routing information for the routes that are to be imported into BGP. Default: no default	
	INTerface	Imports interface routes.
	OSPF	Imports OSPF routes.
	RIP	Imports RIP or RIP2 routes.
	STAtic	Imports statically configured routes.
ROUTEMap	The route map used to filter the routes imported into BGP and to set attributes for the routes as advertised by BGP. The *routemap* is the name of the appropriate pre-existing map. The route map can **match** on origin, next hop, prefix list or tag, and can use any of the **set** parameters. Default: no route map (routes are not filtered and attributes are not set)	

Examples

To import OSPF routes into BGP and use the route map *ospf_bgp_map* to filter and set attributes, use the command:

```
add bgp imp=ospf routem=ospf_bgp_map
```

Related Commands

add ip routemap in Chapter 28, Filtering IP Routes
delete bgp import
enable bgp autosummary
set bgp import
show bgp import

add bgp network

Syntax

```
ADD BGP NETwork=prefix[/0..32] [MASK=mask]
    [ROUTEMap=routemap]
```

Description

This command adds a network to the list of networks that the router can advertise to remote BGP peers. You can also optionally specify a route map to set attributes on the routes sent.

Statically defining a BGP network with this command does not cause the router to advertise the network immediately—the router does not know the next hop for the network. Defining a BGP network informs BGP that if the router learns a route to the network by a non-BGP means, for example statically or from OSPF, then BGP should advertise the network.

If automatic summarising is turned off, BGP advertises the network if it finds a route in the router's RIB to that network. If automatic summarising is turned on, BGP advertises the network if it finds a route in the router's RIB to that network or to any subnet within that network. For more information, see "How to Advertise as Few Routes as Possible" on page 27-30.

Parameter	Description
NETwork	The network to add to the list of networks that can be advertised. This is expressed as the base IP address of the network, in dotted decimal notation, optionally followed by a "/" character and the number of bits in the network mask. If you do not specify the CIDR mask, the router uses the value from the **mask** parameter, if present, or otherwise the natural mask for the network, based on whether it is a class A, B, or C network. Default: no default
MASK	The network mask for the network. This parameter is provided for compatibility with other router commands that specify an IP address and mask; we recommend that you instead specify the mask in the **network** parameter. If you specify a mask in this parameter and the **network** parameter, an error results unless the two masks agree. Default: The natural mask for the network, based on whether it is a class A, B, or C network
ROUTEMap	The route map used to set attributes on routes that are sent in the BGP update messages that advertise this network. The *routemap* is the name of the appropriate pre-existing map. The route map can **match** on origin, next hop, prefix list or tag, and can use any of the **set** parameters. Default: no route map (attributes are not set)

Examples

To add the network 192.169.2.0 to the list of networks advertised by BGP and to use the route map "normal", use the command:

```
add bgp net=192.169.2.0/24 routem=normal
```

Related Commands

delete bgp network
enable bgp autosummary
show bgp network

add bgp peer

Syntax

```
ADD BGP PEer=ipadd REMoteas=1..65534
   [AUthentication={MD5|NONE}]
   [CAPAbilitymatching={STrict|LOOSE}] [CLIEnt={NO|YES}]
   [CONnectretry={DEFault|0..4294967295}]
   [DEFaultoriginate={NO|YES}]
   [DESCription={NONE|description}]
   [EHOps={DEFault|1..255}] [FASTFallover={NO|YES}]
   [HOLdtime={DEFault|0|3..65535}]
   [INFilter={NONE|prefixlist-name}]
   [INPathfilter={NONE|1..99}]
   [INRoutemap={NONE|routemap}]
   [KEEpalive={DEFault|1..21845}]
   [LOCal={NONE|1..15}] [MAXPREFIX={OFF|1..4294967295}]
   [MAXPREFIXAction={Terminate|Warning}]
   [MINAsoriginated={DEFault|0..3600}]
   [MINRouteadvert={DEFault|0..3600}]
   [NEXthopself={NO|YES}]
   [OUTFilter={NONE|prefixlist-name}]
   [OUTPathfilter={NONE|1..99}]
   [OUTRoutemap={NONE|routemap}] [PASSword=password]
   [PRIVateasfilter={NO|YES}] [SENdcommunity={NO|YES}]

ADD BGP PEer=ipadd POLICYTemplate=1..30 REMoteas=1..65534
   [AUthentication={MD5|NONE}]
   [CAPAbilitymatching={STrict|LOOSE}]
   [DEFaultoriginate={NO|YES}]
   [DESCription={NONE|description}]
   [EHOps={DEFault|1..255}] [FASTFallover={NO|YES}]
   [PASSword=password]
```

Description

This command adds a BGP peer to the router. This command adds the peer in the disabled state; the router does not attempt to communicate with the peer until you enter the **enable bgp peer** command on page 27-81. This lets you fully configure the peer entry before communicating with it.

Parameter	Description
PEer	The IP address of the new peer, in dotted decimal notation. This address should be the address that this router uses when communicating with the peer; that is, the address of the interface on the peer that is closest to this router. Default: no default
REMoteas	The remote Autonomous System to which this peer belongs. If the remote AS number is the same as this router's AS number, the peer is an internal BGP (IBGP) peer. If the remote AS number is different from this router's AS number, the peer is an external BGP (EBGP) peer. If the remote AS numbers are different but the routers have the same confederation peer, the peer is a confederation BGP peer. The AS number is assigned by the IANA. Default: no default

Parameter (cont)	**Description** (cont)	
AUthentication	Whether to use MD5 authentication for the BGP peer. If you specify **md5**, you must also specify **password**. Default: **none**	
	MD5	An MD5 digest is added to every BGP packet sent over the TCP connection and is authenticated at the other end. If any part of the digest cannot be verified, the packet is dropped with no response sent.
	NONE	The BGP session is not authenticated.
CAPAbilitymatching	Whether the local BGP speaker (the router) strictly or loosely compares this remote peer's capabilities with its own capabilities. Default: **loose**	
	STrict	The local speaker only establishes a session to the peer if the local speaker supports all the capabilities that the remote peer advertises in its Open message.
	LOOSE	The local speaker and the peer establish a session as long as the remote peer supports at least the IPv4 address family and IPv4 unicast capabilities.
CLIEnt	Whether the peer is a client of the router when the router is a route reflector (RR). RRs selectively advertise routes they learn from their IBGP peers to their other IBGP peers. The router is a route reflector if it has at least one client peer, meaning that **client** is **yes** for at least one of its peers. For more information about route reflection, and client and non-client peers, see "How to Improve IBGP Scalability" on page 27-33. Route reflection is valid for IBGP peers, so **client=yes** is valid only when the local ASN and the remote ASN are the same. Default: **no**	
	NO	The peer is a non-client peer of the RR. When the RR receives a route from a non-client peer, the RR reflects the route only to its client peers, not to any other non-client peers it has.
	YES	The peer is a client peer of the RR. When the RR receives a route from a client peer, the RR reflects the route to all its peers, both client and non-client.
CONnectretry	The time interval between attempts to establish a BGP connection to the peer, in seconds. Default: **120**	
	0	The router does not repeat an attempt to establish a BGP connection.
	1..4294967295	The router waits the specified number of seconds between attempts.
	DEFault	The router waits 120 seconds between attempts.

Parameter (cont)	**Description** (cont)
DEFaultoriginate	Whether to advertise the default route (0.0.0.0/0) to this peer, when the router's BGP routing table contains the default route. To advertise the default route, you need to do all of the following: • set this parameter to yes • create the default route on the router (or the router needs to learn it from another routing source) • configure BGP with an import or network entry that includes the default route • import the default route into the BGP routing table, by using the **enable bgp defaultoriginate** command on page 27-80 Default: **no** (the default route is not propagated from the router's BGP routing table to the peer's RIB)
DESCription	A description of the peer, which has no effect on its operation. A string 1 to 63 characters long. All printable characters are valid except the question mark and double quotes. If *description* contains spaces, the string must be in double quotes. Default:**none**
EHOps	The number of hops put in the *TTL* (Time To Live) field of BGP messages for external BGP. Normally, EBGP requires that BGP peers be connected to a common network, which means they are separated by a single hop. Setting **ehops** to a value greater than 1 indicates that multihop EBGP is allowed. Default: **1**
	1..255 — Number of hops put into the TTL field.
	DEFault — Number of hops put in the TTL field is 1.
FASTfallover	Whether fast fallover is enabled on the link to the peer. If fast fallover is enabled, the peer session is reset as soon as the interface that supports the session goes down. If fast fallover is disabled, the session is reset only when its keepalive timer expires. Default: **no** (fast fallover is disabled)
HOLdtime	The value in seconds that this router proposes for the time interval between reception of keepalive and/or update messages from this peer. The actual hold time used on a peer connection is negotiated when the connection is opened, as the lower of the hold times proposed. Default: **90**
	0 — This router proposes not to have a hold time on this BGP connection.
	3..65535 — This router proposes the specified number of seconds as hold time.
	DEFault — This router proposes a hold time of 90 seconds.

Parameter (cont)	**Description** (cont)
INFilter	The prefix list that filters update messages that the router receives from this peer. If a prefix matches a prefix in the prefix list, BGP rejects that route. Otherwise, it accepts the route. The prefix list must already exist. To create a prefix list, use the **add ip prefixlist** command on page 28-45 of Chapter 28, Filtering IP Routes. If you specify more than one of **inpathfilter**, **infilter** and **inroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the prefix, so a prefix is only included if all the applied filters result in it being included. Default: **none**
INPathfilter	The AS path list that filters the BGP update messages from this peer. You can use an AS path list to exclude or include update messages that have traversed particular ASs or paths. If the path list does not already exist, it is created. To create a path list and/or add entries to it, use the **add ip aspathlist** command on page 28-41 of Chapter 28, Filtering IP Routes. If you specify more than one of **inpathfilter**, **infilter** and **inroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**
INRoutemap	The route map that filters and/or modifies prefixes from this peer. You can use a route map to include or exclude update messages or a subset of an update message's routes, on the basis of a range of BGP attributes, and/or to modify attributes. The route map must already exist. To create a route map use the **add ip routemap** command on page 28-50 of Chapter 28, Filtering IP Routes. If you specify more than one of **inpathfilter**, **infilter** and **inroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**

Parameter (cont)	**Description** (cont)	
KEEpalive	The time in seconds that this router would prefer to leave between keepalive messages to this peer. This time should be one third of the **holdtime** parameter. The actual value used for the keep alive interval is determined once the BGP connection is opened, because the hold time interval is calculated as part of the BGP connection opening. The actual keep alive interval is calculated so that the following ratios are the same: configured keep alive interval: configured hold time interval actual keep alive interval: negotiated hold time interval. If the hold time is negotiated at 0 seconds, then the keep alive interval is also 0 seconds, and keepalive messages are not sent. Default: one third of **holdtime**	
	1..21845	This router prefers the specified number of seconds as keepalive interval.
	DEFault	This router prefers a keepalive interval of one third the hold time.
LOCal	The local interface. In certain circumstances, the router uses this address as the source for BGP packets it generates and sends to this BGP peer. For a description of when the router uses the local interface, see "How to Set the IP Address that Identifies the Router" on page 27-40. Default: **none**	
MAXPREFIX	The maximum number of network prefixes that the router expects to receive from this peer. This parameter provides a safety mechanism in case the peer sends more prefixes than you might normally expect to receive. Default: **off**	
	1..4294967295	The maximum number of prefixes the router expects to receive from this peer. Once this number is exceeded, the action you specify in **maxprefixaction** is carried out.
	OFF	No maximum prefix checking.
MAXPREFIXAction	The action to take when a peer has sent a number of prefixes that exceeds the number specified by **maxprefix**. Default: **warning**	
	Warning	The router logs warnings when the maximum number of prefixes is exceeded.
	Terminate	The router resets the peer connections and logs warnings.
MINAsoriginated	The minimum time in seconds between advertisements, from the router to this peer, of routes that originate in the router's autonomous system. Default: **15**	
	0..3600	Interval is the specified number of seconds.
	DEFault	Interval is 15 seconds.

Parameter (cont)	**Description** (cont)	
MINRouteadvert	The minimum time in seconds between advertisements, from the router to this peer, of routes that originate outside the router's autonomous system. Default: **30**	
	0..3600	The interval is the specified number of seconds.
	DEFault	The interval is 30 seconds.
NEXthopself	Whether this router advertises to this peer that the next hop for all routes is itself. Default: **no**	
	YES	All updates that the router sends to this peer specify this router as the next hop.
	NO	The next hop is specified as described in RFC 1771.
OUTFilter	The prefix list that filters update messages that the router sends to this peer. If a prefix matches a prefix in the prefix list, BGP removes that route from the update message. Otherwise, it leaves the route in the update message and therefore advertises it to the peer. The prefix list must already exist. To create a prefix list, use the **add ip prefixlist** command on page 28-45 of Chapter 28, Filtering IP Routes. If you specify more than one of **outpathfilter**, **outfilter** and **outroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the prefix, so a prefix is only included if all the applied filters result in it being included. Default: **none**	
OUTPathfilter	The AS path list that filters the BGP update messages sent to this peer. You can use an AS path list to exclude or include update messages that have traversed particular ASs or paths. If the path list does not already exist, it is created. To create a path list and/or add entries to it, use the **add ip aspathlist** command on page 28-41 of Chapter 28, Filtering IP Routes. If you specify more than one of **outpathfilter**, **outfilter** and **outroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**	

Parameter (cont)	**Description** (cont)
OUTRoutemap	The route map that filters and/or modifies prefixes sent to this peer. You can use a route map to include or exclude update messages or a subset of an update message's routes, on the basis of a range of BGP attributes, and/or to modify attributes. The route map must already exist. To create a route map use the **add ip routemap** command on page 28-50 of Chapter 28, Filtering IP Routes. If you specify more than one of **outpathfilter**, **outfilter** and **outroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**
PASSword	The key used by the authentication algorithm. Two BGP peers can only communicate with each other if they have the same key. *password* is a character string from 1 to 80 characters long. All printable characters are valid except the question mark and double quotes. If *password* contains spaces, it must be in double quotes. Only valid if **authentication=md5** Default: no default
POLICYTemplate	The ID number of the peer policy template that applies to this peer. The specified policy template must already exist. To create a template, use the **add bgp peertemplate** command on page 27-60. You can only specify **remoteas**, **description**, **authentication**, **password**, **fastfallover,** and **ehops** at the same time as **policytemplate**. The template provides all other configuration values.
PRIVateasfilter	Whether private AS numbers (from 64512 to 65535) are stripped from the AS PATH attribute on update messages the router sends to the peer. Default: **no**
SENdcommunity	Whether the router includes the community attribute in update messages that it sends to this peer. Default: **no** YES — The community attribute is set in update messages to this peer. To set the value of the community attribute, create a route map with a **set** clause to set the community, and use the **outroutemap** parameter to apply it to update messages to this peer. To create a route map use the **add ip routemap** command on page 28-50 of Chapter 28, Filtering IP Routes. NO — The community attribute is not set in update messages to this peer, even if it is set in the route map used by the peer.

Examples To add a BGP peer whose IP address is 192.168.1.1 and whose AS number is 54321, use the command:

```
add bgp pe=192.168.1.1 rem=54321 desc="test remote bgp peer"
```

Related Commands

add ip aspathlist in Chapter 28, Filtering IP Routes
add ip filter in Chapter 22, Internet Protocol (IP)
add ip routemap in Chapter 28, Filtering IP Routes
delete bgp peer
disable bgp peer
enable bgp peer
reset bgp peer
set bgp peer
show bgp peer

add bgp peertemplate

Syntax

```
ADD BGP PEERTemplate=1..30 [CLIEnt={NO|YES}]
   [CONnectretry={DEFault|0..4294967295}]
   [DESCription={NONE|description}]
   [HOLdtime={DEFault|0|3..65535}]
   [INFilter={NONE|prefixlist-name}]
   [INPathfilter={NONE|1..99}]
   [INRoutemap={NONE|routemap}]
   [KEEpalive={DEFault|1..21845}] [LOCal={NONE|1..15}]
   [MAXPREFIX={OFF|1..4294967295}]
   [MAXPREFIXAction={Terminate|Warning}]
   [MINAsoriginated={DEFault|0..3600}]
   [MINRouteadvert={DEFault|0..3600}]
   [NEXthopself={NO|YES}]
   [OUTFilter={NONE|prefixlist-name}]
   [OUTPathfilter={NONE|1..99}]
   [OUTRoutemap={NONE|routemap}]
   [PRIVateasfilter={NO|YES}] [SENdcommunity={NO|YES}]
```

Description This command creates a template for use on BGP peers.

Parameter	Description	
PEERTemplate	ID number of the template. Default: no default	
CLIEnt	Whether peers that use the template are clients of the router if the router is a route reflector (RR). RRs selectively advertise routes they learn from their IBGP peers to their other IBGP peers. The router is a route reflector if it has at least one client peer, meaning that **client** is **yes** for at least one of its peers. For more information about route reflection, and client and non-client peers, see "How to Improve IBGP Scalability" on page 27-33. Route reflection is only valid for IBGP peers, so **client=yes** is only valid when the local ASN and the remote ASN are the same. Default: **no**	
	NO	Peers that use the template are non-client peers of the RR. When the RR receives a route from a non-client peer, the RR reflects the route only to its client peers, not to any other non-client peers it has.
	YES	Peers that use the template are client peers of the RR. When the RR receives a route from a client peer, the RR reflects the route to all its peers, both client and non-client.
CONnectretry	The time interval between attempts to establish a BGP connection to peers that use the template, in seconds. Default: **120**	
	0	The router does not repeat an attempt to establish a BGP connection.
	1..4294967295	The router waits the specified number of seconds between attempts.
	DEFault	The router waits 120 seconds between attempts.

Parameter (cont)	**Description** (cont)	
DESCription	A description for the peers that use the template, which has no effect on their operation. A string 1 to 63 characters long. All printable characters are valid except the question mark and double quotes. If *description* contains spaces, the string must be in double quotes. Default: **none**	
HOLdtime	The value in seconds that this router proposes for the time interval between reception of keepalive and/or update messages from peers that use the template. The actual hold time used on a peer connection is negotiated when the connection is opened, and is the lower of the hold times proposed. Default: **90**	
	0	This router proposes not to have a hold time on this BGP connection.
	3..65535	This router proposes the specified number of seconds as hold time.
	DEFault	This router proposes a hold time of 90 seconds.
INFilter	The prefix list that filters update messages that the router receives from peers that use the template. If a prefix matches a prefix in the prefix list, BGP rejects that route. Otherwise, it accepts the route. The prefix list must already exist. To create a prefix list, use the **add ip prefixlist** command on page 28-45 of Chapter 28, Filtering IP Routes. If you specify more than one of **inpathfilter**, **infilter** and **inroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the prefix, so a prefix is only included if all the applied filters result in it being included. Default: **none**	
INPathfilter	The AS path list that filters the BGP update messages from peers that use the template. You can use an AS path list to exclude or include update messages that have traversed particular ASs or paths. If the path list does not already exist, it is created. To create a path list and/or add entries to it, use the **add ip aspathlist** command on page 28-41 of Chapter 28, Filtering IP Routes. If you specify more than one of **inpathfilter**, **infilter** and **inroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**	
INRoutemap	The route map that filters and/or modifies prefixes from peers that use the template. You can use a route map to include or exclude update messages or a subset of an update message's routes, on the basis of a range of BGP attributes, and/or to modify attributes. The route map must already exist. To create a route map, use the **add ip routemap** command on page 28-50 of Chapter 28, Filtering IP Routes. If you specify more than one of **inpathfilter**, **infilter** and **inroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**	

Parameter (cont)	**Description** (cont)	
KEEpalive	The time in seconds that this router would prefer to leave between keepalive messages to peers that use the template. This time should be one third of the **holdtime** parameter. The actual value used for the keep alive interval is determined once the BGP connection is opened, because the hold time interval is calculated as part of the BGP connection opening. The actual keep alive interval is calculated so that the ratio: configured keep alive interval: configured hold time interval is the same as the ratio: actual keep alive interval: negotiated hold time interval. If the hold time is negotiated at 0 seconds, then the keep alive interval is also 0 seconds, and keepalive messages are not sent. Default: one third of **holdtime**	
	1..21845	This router prefers the specified number of seconds as keepalive interval.
	DEFault	This router prefers a keepalive interval of one third the hold time.
LOCal	The local interface. In certain circumstances, the router uses this address as the source for BGP packets it generates and sends to peers that use this template. For a description of when the router uses the local interface (see "How to Set the IP Address that Identifies the Router" on page 27-40). Default: **none**	
MAXPREFIX	The maximum number of network prefixes that the router expects to receive from peers that use the template. This parameter provides a safety mechanism in case the peer sends more prefixes than you might normally expect to receive. Default: **off**	
	1..4294967295	The maximum number of prefixes the router expects to receive from peers that use the template. Once this number is exceeded, the action you specify in **maxprefixaction** is carried out.
	OFF	No maximum prefix checking.
MAXPREFIXAction	The action to take when a peer has sent a number of prefixes that exceeds the number specified by **maxprefix**. Default: **warning**	
	Warning	The router logs warnings when the maximum number of prefixes is exceeded.
	Terminate	The router resets the peer connections and logs warnings.
MINAsoriginated	The minimum time in seconds between advertisements, from the router to peers that use the template, of routes that originate in the router's autonomous system. Default: **15**	
	0..3600	The interval is the specified number of seconds.
	DEFault	The interval is 15 seconds.

Parameter (cont)	**Description** (cont)	
MINRouteadvert	The minimum time in seconds between advertisements, from the router to peers that use the template, of routes that originate outside the router's autonomous system. Default: **30**	
	0..3600	The interval is the specified number of seconds.
	DEFault	The interval is 30 seconds.
NEXthopself	Whether this router advertises to peers that use the template that the next hop for all routes is itself. Default: **no**	
	YES	All updates that the router sends to peers that use this template specify this router as the next hop.
	NO	The next hop is specified as described in RFC 1771.
OUTFilter	The prefix list that filters update messages that the router sends to peers that use this template. If a prefix matches a prefix in the prefix list, BGP removes that route from the update message. Otherwise, it leaves the route in the update message and therefore advertises it to the peer. The prefix list must already exist. To create a prefix list, use the **add ip prefixlist** command on page 28-45 of Chapter 28, Filtering IP Routes. If you specify more than one of **outpathfilter**, **outfilter** and **outroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the prefix, so a prefix is only included if all the applied filters result in it being included. Default: **none**	
OUTPathfilter	The AS path list that filters the BGP update messages sent to peers that use this template. You can use an AS path list to exclude or include update messages that have traversed particular ASs or paths. If the path list does not already exist, it is created. To create a path list and/or add entries to it, use the **add ip aspathlist** command on page 28-41 of Chapter 28, Filtering IP Routes. If you specify more than one of **outpathfilter**, **outfilter** and **outroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**	
OUTRoutemap	The route map that filters and/or modifies prefixes sent to peers that use this template. You can use a route map to include or exclude update messages or a subset of an update message's routes, on the basis of a range of BGP attributes, and/or to modify attributes. The route map must already exist. To create a route map, use the **add ip routemap** command on page 28-50 of Chapter 28, Filtering IP Routes. If you specify more than one of **outpathfilter**, **outfilter** and **outroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**	

Parameter (cont)	**Description** (cont)	
PRIVateasfilter	Whether private AS numbers (from 64512 to 65535) are stripped from the AS PATH attribute on update messages the router sends to peers that use this template. Default: **no**	
SENdcommunity	Whether the router includes the community attribute in update messages that it sends to peers that use this template. Default: no	
	YES	The community attribute is set in update messages to peers that use this template. To set the value of the community attribute, create a route map with a **set** clause to set the community, and use the **outroutemap** parameter to apply it to update messages to peers that use this template. To create a route map use the **add ip routemap** command on page 28-50 of Chapter 28, Filtering IP Routes.
	NO	The community attribute is not set in update messages to this peer, even if it is set in the route map used by the peers that use this template.

Examples

To create a new peer policy template with a hold time of 30 seconds, and assign it to a peer, use the commands:

```
add bgp peert=1 hol=30
add bgp pe=192.168.1.0/24 policyt=1
```

Related Commands

add bgp peer
set bgp peer
set bgp peertemplate
show bgp peer
show bgp peertemplate

create bgp damping parameterset

Syntax

```
CREate BGP DAMping PARameterset=1..100
   [DESCription=description]
   [SUPpression={DEFault|1..20000}]
   [REUse={DEFault|1..20000}] [HALflife={DEFault|1..45}]
   [MAXhold={DEFault|1..8}]
```

Description

This command creates a parameter set for route flap damping.

If route flap damping is currently enabled as a whole, the new parameter set is enabled. However, the new parameter set is disabled if route flap damping is currently disabled or only particular parameter sets are enabled.

Parameter	Description	
PARameterset	A unique ID number to identify the parameter set. Default: no default	
DESCription	A description of the parameter set, which has no effect on its operation. A string 1 to 63 characters long. All printable characters are valid except the question mark and double quotes. If *description* contains spaces, the string must be in double quotes. Default: no description, but the **show bgp damping** command displays "<Parameterset*n*>" where *n* is the number of the parameter set	
SUPpression	A Figure of Merit (FoM) value, which indicates route stability. When a route's FoM exceeds this threshold, the route is suppressed. **Suppression** must be greater than or equal to **reuse**. If **suppression** is less than 1000, a route is suppressed when it becomes unreachable for the first time. Default: **2000**	
	1..20000	The route is suppressed once its FoM exceeds this value.
	DEFault	The route is suppressed once its FoM exceeds 2000.
REUse	A Figure of Merit (FoM) value, which indicates route stability. Once a route is suppressed, it remains suppressed until its FoM falls below this threshold. **Reuse** must not exceed **suppression**. Default: **750**	
	1..20000	The route becomes available again once its FoM drops to this value.
	DEFault	The route becomes available again once its FoM drops to 750.
HALflife	The interval in minutes during which the route's FoM will halve if the route remains stable. For example, if **halflife** is 15, the FoM of a stable route reduces by 50% over a 15 minute period, 75% over a 30 minute period, and so on. Default: **15**	
	1..45	The FoM of a stable route halves in this number of minutes.
	DEFault	The FoM of a stable route halves in 15 minutes.

Parameter (cont)	**Description** (cont)	
MAXhold	When multiplied by **halflife**, gives the maximum time in minutes for which a suppressed route must remain stable in order to become unsuppressed. The lowest **maxhold** value of 1 gives a maximum suppression time of 1 x **halflife**, and the highest **maxhold** value of 8 gives a maximum suppression time of 8 x **halflife**. For example, if **halflife** is 15 and **maxhold** is 4, the route is unsuppressed after 60 minutes of stability even if its FoM still exceeds **reuse**. Default: **4**	
	1..8	The **halflife** is multiplied by this value to give the maximum suppression time.
	DEFault	The **halflife** is multiplied by 4 to give the maximum suppression time.

Examples

To create BGP route flap damping parameter set 3 with a halflife of 5 minutes and a suppression threshold of 3000, use the command:

```
create bgp damping parameterset=3 halflife=5 suppression=3
```

This set is more tolerant of route instability than the default.

Related Commands

add bgp peer
disable bgp damping
enable bgp damping
show bgp damping
show bgp damping routes

delete bgp aggregate

Syntax `DELete BGP AGGregate=`*`prefix`* `[MASK=`*`ipadd`*`]`

Description This command deletes an aggregate entry from BGP. BGP no longer advertises the aggregate entry. If the aggregate entry is currently in the BGP route table, BGP also sends an update message to all peers to withdraw the route. Associated aggregate suppressed routes are then reconsidered for advertisement.

Parameter	Description
AGGregate	The network prefix of the aggregate entry to delete. This is expressed as the base IP address of the network, in dotted decimal notation, optionally followed by a "/" character and the number of bits in the network mask. If you do not specify the CIDR mask, the router uses the value from the **mask** parameter, if present, or otherwise the natural mask for the network, based on whether it is a class A, B, or C network. Default: no default
MASK	The network mask for the aggregate entry. This parameter is provided for compatibility with other router commands that specify an IP address and mask; we recommend that you instead specify the mask in the **aggregate** parameter. If you specify a mask in this parameter and the **aggregate** parameter, an error results unless the two masks agree. Default: The natural mask for the network, based on whether it is a class A, B, or C network.

Examples To delete the aggregate entry for the network 192.168.8.0/21, use the command:

```
del bgp agg=192.168.8.0/21
```

Related Commands

add bgp aggregate
set bgp aggregate
show bgp aggregate

delete bgp confederationpeer

Syntax `DELete BGP CONFEDerationpeer=1..65534`

Description This command deletes an Autonomous System from the AS confederation to which this router belongs. An AS confederation is a group of Autonomous Systems that communicate between themselves using confederation BGP, and communicate to Autonomous Systems outside the confederation as if they were a single Autonomous System. For more information about AS confederations, see "AS Confederations" .

The **confederationpeer** parameter specifies the number of an Autonomous System that is no longer to be treated as one of the Autonomous Systems in the confederation. The specified AS number must be an already existing AS confederation peer.

Examples To remove AS 60003 from the AS confederation to which this router belongs, use the command:

```
del bgp confed=60003
```

Related Commands
add bgp confederationpeer
set bgp
show bgp confederation

delete bgp import

Syntax `DELete BGP IMPort={INTerface|OSPF|RIP|STAtic}`

Description This command deletes an import entry from BGP. Routes from the source specified are no longer imported into the BGP route table. Routes already in the BGP route table are removed.

Parameter	Description	
IMPort	The source of routing information for the routes that are no longer to be imported into BGP. Default: no default	
	INTerface	Stops the import of interface routes.
	OSPF	Stops the import of OSPF routes.
	RIP	Stops the import of RIP or RIP2 routes.
	STAtic	Stops the import of statically configured routes.

Examples To stop importing OSPF routes into BGP, use the command:

```
del bgp imp=ospf
```

Related Commands
add bgp import
set bgp import
show bgp import

delete bgp network

Syntax `DELete BGP NETwork=`*`prefix`* `[MASK=`*`ipadd`*`]`

Description This command deletes a network from the list of networks that the router can advertise to remote BGP peers. If the router had previously advertised the route, BGP sends an update message to all peers to withdraw the network.

Parameter	Description
NETwork	The network to remove from the list of networks that can be advertised. This is expressed as the base IP address of the network, in dotted decimal notation, optionally followed by a "/" character and the number of bits in the network mask. If you do not specify the CIDR mask, the router uses the value from the **mask** parameter, if present, or otherwise the natural mask for the network, based on whether it is a class A, B, or C network. Default: no default
MASK	The network mask for the network. This parameter is provided for compatibility with other router commands that specify an IP address and mask; we recommend that you instead specify the mask in the **network** parameter. If you specify a mask in this parameter and the **network** parameter, an error results unless the two masks agree. Default: The natural mask for the network, based on whether it is a class A, B, or C network

Examples To delete the network 192.169.2.0 from the list of networks advertised by BGP, use the command:

```
del bgp net=192.169.2.0/24
```

Related Commands
add bgp network
show bgp network

delete bgp peer

Syntax `DELete BGP PEer=`*`ipadd`*

Description This command deletes a BGP peer from the router. The BGP peer must be in a disabled state: either never enabled, or previously enabled and subsequently disabled with the **disable bgp peer** command.

The **peer** parameter specifies the IP address of the peer to be deleted, in dotted decimal notation. The peer must be an existing BGP peer on this router.

Examples To delete a BGP peer whose IP address is 192.168.1.1, use the command:

```
del bgp pe=192.168.1.1
```

Related Commands
disable bgp peer
show bgp peer

delete bgp peertemplate

Syntax `DELete BGP PEERTemplate=1..30`

Description This command deletes an existing BGP peer policy template from the router. All peers that have been assigned the specified peer template receive their own copies of the current peer template settings. You can subsequently modify these peers.

The **peertemplate** parameter specifies the ID number of the template to be deleted.

Examples To delete BGP peer template 1, use the command:

```
del bgp peert=1
```

Related Commands **add bgp peertemplate**
show bgp peer

destroy bgp damping parameterset

Syntax `DESTroy BGP DAMping PARameterset={ALL|1..100}`

Description This command removes the specified BGP route flap damping parameter set from the group of available parameter sets. You can destroy a parameter set only if BGP damping is disabled for that parameter set or as a whole.

The **parameterset** parameter specifies the parameter set to destroy. If you specify **all**, all user-defined parameter sets are destroyed.

The default parameter set, numbered 0, cannot be destroyed, but you can modify its settings by using the command **set bgp damping parameterset=0**.

Example To destroy parameter set 3, use the command:

```
destroy bgp damping parameterset=3
```

Related Commands **add bgp peer**
disable bgp damping
enable bgp damping
show bgp damping
show bgp damping routes

disable bgp autosoftupdate

Syntax `DISable BGP AUTOsoftupdate`

Description This command disables automatic updating of modified BGP peers. Changes to a peer take effect only when the peer next receives or sends an update message, unless you manually trigger the update with the **reset bgp peer soft** command. Automatic updating is disabled by default.

Examples To disable automatic updating, use the command:

```
dis bgp auto
```

Related Commands
reset bgp peer
reset bgp peer soft
set bgp peer
show bgp peer

disable bgp autosummary

Syntax `DISable BGP AUTOSUmmary`

Description This command stops the router from automatically summarising locally originated or imported subnet routes into a single route.

Automatic summarisation is disabled by default.

Example To disable automatic summarisation, use the command:

```
dis bgp autosu
```

Related Commands
add bgp import
add bgp network
enable bgp autosummary
show bgp
show bgp route

disable bgp backoff

Syntax `DISable BGP BACkoff`

Description This command stops BGP backoff. BGP backoff delays BGP processing when the system memory utilisation is high.

BGP backoff is disabled by default, however it automatically enables the first time a peer is added.

Example To disable BGP backoff, use the command:

```
dis bgp bac
```

Related Commands **enable bgp backoff**
set bgp backoff
show bgp backoff

disable bgp damping

Syntax `DISable BGP DAMping [PARameterset={ALL|0..100}]`

Description This command disables monitoring and suppression of flapping BGP routes by disabling BGP route flap damping for one or all parameter sets. This command clears route stability history information and may be used to turn off route flap damping temporarily for configuration changes.

The **parameterset** parameter specifies the parameter set to disable. If you do not specify **parameterset**, all route flap damping is disabled. If you specify **parameterset**, only that parameter set is disabled, not all route flap damping, unless you specify the last enabled parameter set. In that case the feature is disabled, which is equivalent to not specifying a parameter set.

Examples To disable BGP route flap damping for all enabled parameter sets and the feature, use the command:

```
disable bgp damping
```

To disable BGP route flap damping for parameter set 3 only, use the command:

```
disable bgp damping parameterset=3
```

If parameter set 3 is the last enabled parameter set, the feature is disabled.

Related Commands
add bgp peer
enable bgp damping
show bgp damping
show bgp damping routes

disable bgp debug

Syntax

```
DISable BGP DEBug[={ALL|DAMping|MSG|STAte|UPdate}[,...]]
   [PEer=ipadd]
```

Description

This command disables one or more forms of BGP debugging, optionally for a given BGP peer.

You can direct BGP debugging to only one manager device at a time. This means that if someone is debugging BGP on another terminal device, you cannot enable debugging on the current terminal device. However, you can use this command to disable debugging for the other device, and then enable debugging for the current device.

Parameter	Description	
DEBug	Debugging options to disable specified as: • one option • a comma-separated list Default: **all**	
	ALL	All debugging options.
	DAMping	Messages to reflect BGP route flap damping state changes and events. Events include routes getting penalised for flapping, route state transition to suppressed and to reuse, routes becoming reachable and routes becoming unreachable.
	MSG	Message reception and transmission. There are four message types: open, update, keepalive, and notify. The output of the debug messages consists of the timestamp, the direction of the message, incoming or outgoing, the IP address of the peer, the message type and the details. For open, keepalive, and notify messages, the details consist of the decoded contents of the packet. For update messages, the details consist of the lengths of the withdrawn routes and NRLI fields, and a complete decode of the path attributes.
	STAte	State machine events and transitions. The output of the debug messages consists of the timestamp, the IP address of the peer, the event that causes the state change, and the old and new states.
PEer	The IP address of the peer for which debugging is no longer required, in dotted decimal notation. The peer must be an existing BGP peer on this router. Default: no default (debugging is turned off for all BGP peers)	

Examples

To disable all debugging for all BGP peers, use the command:

```
dis bgp deb
```

Related Commands

enable bgp debug
disable debug active in Chapter 4, Configuring and Monitoring the System
show debug active in Chapter 4, Configuring and Monitoring the System

disable bgp defaultoriginate

Syntax `DISable BGP DEFaultoriginate`

Description This command prevents BGP from importing the default route (0.0.0.0/0) into its routing table. This command over-rides other import options, so BGP does not import the default route even when it is configured with an import or network entry that includes the default route.

This feature is disabled by default. Therefore, by default BGP excludes the default route.

Example To prevent BGP from importing the default route, use the command:

```
dis bgp def
```

Related Commands
enable bgp defaultoriginate
add bgp network
show bgp

disable bgp peer

Syntax `DISable BGP PEer={ALL|`*ipadd*`}`

Description This command disables a given BGP peer, or all BGP peers. The router destroys its TCP connection to the peer, and the associated BGP session. The router and the peer withdraw any routes they learned during that session.

The **peer** parameter specifies the IP address of the peer to disable, in dotted decimal notation. If you specify **all**, all BGP peers are disabled.

Examples To disable the BGP peer 192.168.1.1, use the command:

```
dis bgp pe=192.168.1.1
```

Related Commands
enable bgp peer
show bgp peer

enable bgp autosoftupdate

Syntax `ENAble BGP AUTOsoftupdate`

Description This command enables the router to automatically update BGP peers after you modify them. Automatic updating is disabled by default.

Examples To enable automatic updating, use the command:

```
ena bgp auto
```

Related Commands
reset bgp peer
set bgp peer
show bgp peer

enable bgp autosummary

Syntax `ENAble BGP AUTOSUmmary`

Description This command enables the router to automatically summarise locally originated or imported subnet routes. When automatic summarising is enabled, the router summarises routes that are under the same classful network to a single route of the classful network. The router imports and advertises the summary route instead. Automatic summarisation is disabled by default.

For more information, see "How to Advertise as Few Routes as Possible" on page 27-30.

Example To enable automatic summarisation, use the command:

```
ena bgp autosu
```

Related Commands
add bgp import
add bgp network
disable bgp autosummary
show bgp
show bgp route

enable bgp backoff

Syntax `ENAble BGP BACkoff`

Description This command allows BGP backoff. BGP backoff delays BGP processing when the system memory utilisation is high.

BGP backoff is disabled by default, however it automatically enables the first time a peer is added.

Example To enable BGP backoff, use the command:

```
ena bgp bac
```

Related Commands
disable bgp backoff
set bgp backoff
show bgp backoff

enable bgp damping

Syntax `ENAble BGP DAMping [PARameterset={ALL|0..100}]`

Description This command enables monitoring and suppression of flapping BGP routes through route flap damping. Route flap damping is disabled by default. Use this command to enable it on:

- all routes by using the command:

  ```
  enable bgp damping
  ```

  This enables damping itself, and all parameter sets. If you associate particular parameter sets with particular peers, the router applies those parameter sets to update messages from those peers. For all other update messages, the router uses the default parameter set (Figure 27-11 on page 27-28).

- some or all routes from a specific BGP peer by using the command:

  ```
  enable bgp damping parameterset=1..100
  ```

  You also need to associate the parameter set with the peer by specifying the parameter set in a route map and applying that route map to the peer. See "How to Configure Route Flap Damping" for more information.

- all routes when route flap damping has been previously enabled on only some of the parameter sets by using the command:

  ```
  enable bgp damping parameterset=all
  ```

- all routes that are not associated with a user-defined parameter set by using the command:

  ```
  enable bgp damping parameterset=0
  ```

  The default parameter set is used on these routes.

The **parameterset** parameter specifies the ID number of the parameter set to enable.

Examples To enable BGP route flap damping for all existing parameter sets, use the command:

```
enable bgp damping
```

To enable BGP route flap damping for parameter set 3, use the command:

```
enable bgp damping parameterset=3
```

If some of the parameter sets are currently enabled, enable BGP route flap damping for all other parameter sets by using the command:

```
enable bgp damping parameterset=all
```

Related Commands

add bgp peer
disable bgp damping
show bgp damping
show bgp damping routes

enable bgp debug

Syntax

```
ENAble BGP DEBug={ALL|DAMping|MSG|STAte|UPdate}[,...]
   [PEer=ipadd]
```

Description

This command enables one or more forms of BGP debugging, optionally for a given BGP peer.

You can direct BGP debugging to only one manager device at a time. This means that if someone is debugging BGP on another terminal device, you cannot enable debugging on the current terminal device. However, you can use the **disable bgp debug** command to disable debugging for the other device, and then enable debugging for the current device.

Parameter	Description	
DEBug	Debugging options to enable specified as: • one option • a comma-separated list Default: **all**	
	ALL	All debugging options.
	DAMping	Messages to reflect BGP route flap damping state changes and events. Events include routes getting penalised for flapping, route state transition to suppressed and to reuse, routes becoming reachable and routes becoming unreachable.
	MSG	Message reception and transmission. There are four message types: open, update, keepalive, and notify. The output of the debug messages consists of the timestamp, the direction of the message, incoming or outgoing, the IP address of the peer, the message type and the details. For open, keepalive, and notify messages, the details consist of the decoded contents of the packet. For update messages, the details consist of the lengths of the withdrawn routes and NRLI fields, and a complete decode of the path attributes.
	STAte	State machine events and transitions. The output of the debug messages consists of the timestamp, the IP address of the peer, the event that causes the state change, and the old and new states.
PEer	The IP address of the peer for which debugging is required, in dotted decimal notation. Default: no default (debugging is turned on for all BGP peers)	

Examples

To enable message and state debugging for BGP peer 192.168.1.1, use the command:

```
ena bgp deb=msg,state peer=192.168.1.1
```

Related Commands

disable bgp debug
disable debug active in Chapter 4, Configuring and Monitoring the System
show debug active in Chapter 4, Configuring and Monitoring the System

enable bgp defaultoriginate

Syntax `ENAble BGP DEFaultoriginate`

Description This command enables BGP to import the default route (0.0.0.0/0) into its routing table. You also need to do both of the following:

- create the default route on the router (or the router needs to learn it from another routing source)
- configure BGP with an import or network entry that includes the default route

This feature is disabled by default. Therefore, by default BGP excludes the default route, even when an appropriate import or network entry exists.

To configure an import entry that includes the default route, use the **add bgp import** command on page 27-50 and specify the default route type in the **import** parameter (most often **import=static**).

To configure a network entry that includes the default route, use the **add bgp network** command on page 27-51 and specify **network=0.0.0.0/0**.

This command does not determine whether the router advertises the default route to its BGP peers. You can configure that for each peer, by using the **defaultoriginate** parameter of the **add bgp peer** command on page 27-52.

Note that you do not need to enable this feature if you want to aggregate subnets of 0.0.0.0 into a single network route of 0.0.0.0/0. Instead, create an aggregate entry of 0.0.0.0/0 by using the **add bgp aggregate** command on page 27-47.

Example To enable BGP to import the default route, use the command:

```
ena bgp def
```

Related Commands

add bgp aggregate
add bgp import
add bgp network
add bgp peer
disable bgp defaultoriginate
show bgp

enable bgp peer

Syntax `ENAble BGP PEer={ALL|`*ipadd*`}`

Description This command enables a specific BGP peer or all BGP peers. The router initialises BGP resources, initiates a TCP connection to the peer, and listens for connections initiated by the peer.

The **peer** parameter specifies the IP address of the peer to enable, in dotted decimal notation. The peer must already exist. If you specify **all**, all BGP peers are enabled.

Examples To enable all BGP peers, use the commands:

```
ena bgp pe=all
```

Related Commands
disable bgp peer
reset bgp peer
show bgp peer

purge bgp damping

Syntax `PURge BGP DAMping`

Description This command purges all configuration information relating to BGP route flap damping. All user-defined parameter sets are destroyed. Accumulated route stability history is cleared, and the route flap damping is disabled.

Caution All current configuration and stability history information will be lost.

Example To purge BGP route flap damping, use the command:

```
purge bgp damping
```

Related Commands
create bgp damping parameterset
disable bgp damping
enable bgp damping
reset bgp damping
show bgp damping

reset bgp damping

Syntax `RESET BGP DAMping [PARameterset={ALL|0..100}]`

Description This command clears the BGP route flap damping stability history for all BGP routes, or the routes attached to the specified **parameterset**.

The **parameterset** parameter specifies a parameter set. The stability history of all routes attached to this parameter set is reset. If you do not specify **parameterset**, or if you specify **parameterset=all**, all bgp damping routes are reset.

Example To clear BGP route flap damping stability history for all routes attached to parameter set 3, use the command:

```
reset bgp damping parameterset=3
```

Related Commands
create bgp damping parameterset
disable bgp damping
enable bgp damping
purge bgp damping
show bgp peer

reset bgp peer

Syntax `RESET BGP PEer={ALL|ipadd}`

Description This command resets a specific BGP peer or all BGP peers. This is the equivalent of disabling the peer, then immediately enabling it. The router destroys its TCP connection to the peer and the associated BGP session. The router and the peer withdraw routes they learned from that session. Then the router initialises BGP resources, initiates a TCP connection to the peer, and listens for connections initiated by the peer.

The **peer** parameter specifies the IP address of the peer to reset, in dotted decimal notation. If you specify **all**, all BGP peers are reset.

Examples To reset BGP peer 192.168.1.1, use the command:

```
reset bgp pe=192.168.1.1
```

Related Commands
enable bgp peer
disable bgp peer
show bgp peer

reset bgp peer soft

Syntax `RESET BGP PEer={ALL|`*ipadd*`} SOft={IN|OUt|ALl}`

Description This command updates all BGP peers or a specific one after you have modified the peer. You do not need this command if automatic updating has been enabled with the **enable bgp autosoftupdate** command.

Parameter	Description	
PEer	The peer or peers to update. Default: no default	
	ALL	All peers are updated.
	ipadd	Only the specified peer is updated. The peer is identified by its IP address in dotted decimal notation.
SOft	The direction to update. Default: no default	
	IN	Updates routes that the peer receives. To trigger this update, the peer sends a Route Refresh message to the remote peers from which it receives routes. The Route Refresh message triggers the remote peers to resend a BGP Update message.
	OUt	Updates routes that the peer sends. To reset these, the peer simply sends a BGP Update message to the affected BGP peers.
	ALl	Updates routes the peer sends and receives.

Examples To update BGP peer 192.168.1.1 after you have changed the filter it uses on incoming routes, use the command:

```
reset bgp pe=192.168.1.1 soft=in
```

Related Commands
enable bgp autosoftupdate
enable bgp peer
disable bgp peer
show bgp peer

set bgp

Syntax

```
SET BGP [CLUSter=ipadd] [CONfederationid={NONE|1..65534}]
   [LOCalpref={DEFault|0..4294967295}]
   [MED={NONE|0..4294967294}] [PREFExt={DEFault|1..255}]
   [PREFInt={DEFault|1..255}] [ROuterid=ipadd]
   [SELEction_timer=3..60] [TABlemap[=routemap]
```

Description This command sets global BGP parameters in the router.

Parameter	Description
CLUSter	The cluster ID of the cluster for which the router is a route reflector (RR). **cluster** is only used if the router is performing BGP route reflection (see "How to Improve IBGP Scalability" on page 27-33). By default, the cluster ID is the local BGP identifier. This is sufficient when the router is the only RR in its cluster. However, if the cluster contains multiple RRs, you must give all of the RRs the same cluster ID, which should be the IP address of one of the RRs. Failure to do so may result in routing information loops. Default: the local BGP identifier
CONfederationid	The AS number of the AS confederation to which this router belongs. This AS number is used as the AS number for this router when communicating with BGP peers outside the confederation. A peer is outside the confederation if its AS number is different from this router's AS number, and it is not one of this router's confederation peers. For more information about AS confederations, see "AS Confederations" on page 27-13. You need to specify this parameter if the router has peer relationships with any BGP routers outside the confederation. When you create the peer relationship using the **add bgp peer** command on page 27-52, specify the peer's confederation ID in the **remoteas** parameter. Default: **none**
LOCalpref	The local preference value used in all update messages sent to internal peers when this is not overridden by changes due to the actions in a route map. The local preference is used in internal BGP to decide which BGP route to put into the main routing table. The route with the highest value of local preference is used. Default: **100**
DEFault	The local preference is 100.
0..4294967295	The local preference is the specified number.
MED	The Multi-Exit Discriminator (MED) value that is placed in update messages to external BGP peers from this router when not overridden by the action of a route map. The MED attribute is used by routers in one AS to distinguish between different exit points in the same neighbouring AS. A route with a lower value of MED is used, all other things being equal. For more information about the use of MED in route selection, see "BGP Route Selection" on page 27-8. Default: **none** (no MED attribute is put in update messages)

Parameter (cont)	**Description** (cont)
PREFExt	The route preference BGP gives to routes that it learns from external peers. The router uses routes with a lower preference value before routes with a higher preference. This parameter has been superseded by the **set ip route preference** command on page 22-166 of Chapter 22, Internet Protocol (IP), but is still accepted for backwards compatibility. The **set ip route preference** command allows a wider range of preference values. When you save your configuration using the **create config** command, the router converts the **prefext** parameter to a **set ip route preference** command. Default: **170**
	DEFault — The route preference is 170.
	1..255 — The route preference is the specified number.
PREFInt	The route preference BGP gives to routes that it learns from internal peers. The router uses routes with a lower preference value before routes with a higher preference. This parameter has been superseded by the **set ip route preference** command on page 22-166 of Chapter 22, Internet Protocol (IP), but is still accepted for backwards compatibility. The **set ip route preference** command allows a wider range of preference values. When you save your configuration using the **create config** command, the router converts the **prefint** parameter to a **set ip route preference** command. Default: **170**
	DEFault — The route preference is 170.
	1..255 — The route preference is the specified number.
ROuterid	A 4-byte number that uniquely identifies the router in a network system in certain circumstances, specified as an IP address in dotted decimal notation. For a description of when the router uses the router ID, see "How to Set the IP Address that Identifies the Router" on page 27-40. Default: The default local interface's IP address, if it is configured. Otherwise, the highest interface IP address on the router.
SELEction_timer	The time in seconds that the router waits, after changes to its BGP routing information database, before it determines whether the changes need to be propagated to its BGP peers. By deferring the operation, multiple changes may be able to be aggregated into a single update. Therefore accepting slightly longer convergence times may reduce the BGP messaging overhead. Default: **15** seconds.
TABlemap	A route map for BGP routes to be passed through before installing the route into the routing table. If you specify **tablemap** without specifying a routemap, you clear the **tablemap** setting and the router does not pass routes through a route map before writing them into the routing table. Default: no default (the router does not pass routes through a route map before writing them into the routing table)

Examples To set the preference of routes learned by internal BGP to be better than the preference of routes learned by OSPF (which is 10), use the command:

```
set bgp prefi=9
```

Related Commands

add bgp confederationpeer
delete bgp confederationpeer
show bgp
show bgp confederation

set bgp aggregate

Syntax

```
SET BGP AGGregate=prefix [MASK=ipadd] [SUMmary={NO|YES}]
   [ROUTEMap[=routemap]]
```

Description

This command modifies the parameters of an existing aggregate entry in the BGP route table.

Parameter	Description
AGGregate	The aggregate entry to modify, which is identified by its network prefix. This is expressed as the base IP address of the network, in dotted decimal notation, optionally followed by a "/" character and the number of bits in the network mask. If you do not specify the CIDR mask, the router uses the value from the **mask** parameter, if present, or otherwise the natural mask for the network, based on whether it is a class A, B, or C network. Default: no default
MASK	The network mask for the aggregate entry. This parameter is provided for compatibility with other router commands that specify an IP address and mask; we recommend that you instead specify the mask in the **aggregate** parameter. If you specify a mask in this parameter and the **aggregate** parameter, an error results unless the two masks agree. Default: The natural mask for the network, based on whether it is a class A, B, or C network
SUMmary	Whether the router advertises only the aggregate route, or also the more specific routes that make up the aggregate. Default: **no** **no** The router advertises the more specific routes that make up the aggregate. **yes** The router only advertises the aggregate route. Note that unadvertised routes are still displayed in output of the **show bgp route** command, but are marked with an "s".
ROUTEMap	The route map used to filter the more specific routes that make up the aggregate, or to set attributes for the aggregate route. The *routemap* is the name of the appropriate pre-existing map. Default: no route map (routes are not filtered and attributes are not set)

Examples

To set the route map for the aggregate entry for the network 192.168.8.0/21 to be route map *agg_map1*, use the command:

```
set bgp agg=192.168.8.0/21 routem=agg_map1
```

Related Commands

add bgp aggregate
delete bgp aggregate
show bgp aggregate

set bgp backoff

Syntax

```
SET BGP BACkoff[=20..100] [BASEtime=0..100]
   [CONSecutive=0..1000] [LOW=15..99] [MULtiplier=1..1000]
   [STep=1..1000] [TOTallimit=0..1000]
```

Description

This command configures BGP backoff, which stops BGP processing when system memory is heavily used by another process.

Parameter	Description
BACkoff	The percentage of total system memory use that triggers BGP to back off, from 20 to 100. This must be set higher than the **low** parameter. Default: 95
LOW	The percentage of total system memory use that the router must fall below before BGP backoff will end, from 15 to 99. This must be set lower than the **backoff** parameter. Default: 90
BASEtime	The time value in seconds used to calculate the total backoff time for the first backoff iteration, from 0 to 100. The first backoff time is calculated as: `basetime x multiplier/100` Default: 10
CONSecutive	The number of consecutive backoffs that causes BGP to disable all peers, from 1 to 20. Default: 5
MULtiplier	A multiplier for increasing or decreasing the backoff time at each backoff iteration, from 0 to 1000. The change in backoff time at each step is calculated as: `current backoff time x multiplier/100` Default: 100
STep	The number of backoff iterations after which the backoff time is recalculated. Default: 1
TOTallimit	The total number of backoffs that may occur until all peers are disabled. Default: 0 (no limit)

Examples

To back BGP processing off when total system memory is 90% utilised, and reinstate it when system memory is at 80%, use the command:

```
set bgp bac=90 low=80
```

Related Commands

disable bgp backoff
enable bgp backoff
show bgp backoff
show bgp memlimit

set bgp damping parameterset

Syntax

```
SET BGP DAMping PARameterset=1..100
   [DESCription=description]
   [SUPpression={DEFault|1..20000}]
   [REUse={DEFault|1..20000}] [HALflife={DEFault|1..45}]
   [MAXhold={DEFault|1..8}]
```

Description This command modifies the settings of a parameter set for route flap damping.

Parameter	Description
PARameterset	A unique ID number that identifies the parameter set. Default: no default
DESCription	A description of the parameter set, which has no effect on its operation. A string 1 to 63 characters long. All printable characters are valid except the question mark and double quotes. If *description* contains spaces, the string must be in double quotes. Default: no description, but the **show bgp damping** command displays "<Parameterset*n*>" where *n* is the number of the parameter set
SUPpression	A Figure of Merit (FoM) value, which indicates route stability. When a route's FoM exceeds this threshold, the route is suppressed. **Suppression** must be greater than or equal to **reuse**. If **suppression** is less than 1000, a route is suppressed when it becomes unreachable for the first time. Default: **2000**
1..20000	The route is suppressed once its FoM exceeds this value.
DEFault	The route is suppressed once its FoM exceeds 2000.
REUse	A Figure of Merit (FoM) value, which indicates route stability. Once a route is suppressed, it remains suppressed until its FoM falls below this threshold. **Reuse** must not exceed **suppression**. Default: **750**
1..20000	The route becomes available again once its FoM drops to this value.
DEFault	The route becomes available again once its FoM drops to 750.
HALflife	The time interval, in minutes, within which the route's FoM will halve, if the route remains stable. For example, if **halflife** is 15, the FoM of a stable route reduces by 50% over a 15 minute period, 75% over a 30 minute period, and so on. Default: **15**
1..45	The FoM of a stable route halves in this number of minutes.
DEFault	The FoM of a stable route halves in 15 minutes.

Parameter (cont)	**Description** (cont)	
MAXhold	When multiplied by **halflife**, gives the maximum time in minutes for which a suppressed route must remain stable in order to become unsuppressed. The lowest **maxhold** value of 1 gives a maximum suppression time of 1 x **halflife**, and the highest **maxhold** value of 8 gives a maximum suppression time of 8 x **halflife**. For example, if **halflife** is 15 and **maxhold** is 4, the route is unsuppressed after 60 minutes of stability even if its FoM still exceeds **reuse**. Default: **4**	
	1..8	The **halflife** is multiplied by this value to give the maximum suppression time.
	DEFault	The **halflife** is multiplied by 4 to give the maximum suppression time.

Examples

To set BGP route flap damping parameter set 3 to have a halflife of 5 minutes and a suppression threshold of 3000, use the command:

```
set bgp damping parameterset=3 halflife=5 suppression=3
```

This set is more tolerant of route instability than the default.

Related Commands

add bgp peer
disable bgp damping
enable bgp damping
show bgp damping
show bgp damping routes

set bgp import

Syntax

```
SET BGP IMPort={INTerface|OSPF|RIP|STAtic}
   [ROUTEMap[=routemap]]
```

Description

This command associates a different route map with a BGP import entry, or disassociates a route map. The import entry instructs BGP to import routes from that route source into the BGP route table, and the route map allows filtering of routes and setting of BGP attributes.

Parameter	Description	
IMPort	The BGP import entry to be modified. This must already have been added to BGP by using the **add bgp import** command. Default: no default	
	INTerface	Imports interface routes.
	OSPF	Imports OSPF routes.
	RIP	Imports RIP or RIP2 routes.
	STAtic	Imports statically configured routes.
ROUTEMap	The route map used to filter the routes imported into BGP and to set attributes for the routes as advertised by BGP. The *routemap* is the name of the appropriate pre-existing map. The route map can **match** on origin, next hop, prefix list or tag, and can use any of the **set** parameters. If you specify the **routemap** parameter without specifying a routemap name, the existing route map is removed from the import entry. Default: no route map (routes are not filtered and attributes are not set)	

Examples

To change the route map for the import entry for importing OSPF routes into BGP to route map *ospf_bgp_map1*, use the command:

```
set bgp imp=ospf routem=ospf_bgp_map1
```

Related Commands

add bgp import
delete bgp import
show bgp import

set bgp memlimit

Syntax `SET BGP MEMlimit[=0..100]`

Description This command limits the percentage of system memory available to BGP and IP routing.

The **memlimit** parameter specifies the maximum percentage of system memory that BGP and IP routing combined can use. When the routing exceeds this percentage, the router shuts down BGP peers and drops all routes learnt from the peers. The default is 85%.

Example To limit BGP to 90% of system memory, use the command:

```
set bgp mem=90
```

Related Commands **set bgp backoff**
show bgp memlimit
show bgp memlimit scan
show buffer in Chapter 4, Configuring and Monitoring the System

set bgp peer

Syntax

```
SET BGP PEer=ipadd [AUthentication={MD5|NONE}]
   [CAPAbilitymatching={STrict|LOOSE}] [CLIEnt={NO|YES}]
   [CONnectretry={DEFault|0..4294967295}]
   [DEFaultoriginate={NO|YES}]
   [DESCription={NONE|description}]
   [EHOps={DEFault|1..255}] [FASTFallover={NO|YES}]
   [HOLdtime={DEFault|0|3..65535}]
   [INFilter={NONE|prefixlist-name}]
   [INPathfilter={NONE|1..99}]
   [INRoutemap={NONE|routemap}]
   [KEEpalive={DEFault|1..21845}] [LOCal={NONE|1..15}]
   [MAXPREFIX={OFF|1..4294967295}]
   [MAXPREFIXAction={Terminate|Warning}]
   [MINAsoriginated={DEFault|0..3600}]
   [MINRouteadvert={DEFault|0..3600}]
   [NEXthopself={NO|YES}]
   [OUTFilter={NONE|prefixlist-name}]
   [OUTPathfilter={NONE|1..99}]
   [OUTRoutemap={NONE|routemap}] [PASSword=password]
   [PRIVateasfilter={NO|YES}] [REMoteas=1..65534]
   [SENdcommunity={NO|YES}]

SET BGP PEer=ipadd POLICYTemplate=

SET BGP PEer=ipadd [POLICYTemplate=1..30]
   [AUthentication={MD5|NONE}]
   [CAPAbilitymatching={STrict|LOOSE}]
   [DEFaultoriginate={NO|YES}]
   [DESCription={NONE|description}]
   [EHOps={DEFault|1..255}] [FASTFallover={NO|YES}]
   [PASSword=password] [REMoteas=1..65534]
```

Description

This command modifies parameters for an existing BGP peer in the router.

While any changes you make using the parameters **nexthopself**, **outfilter**, **outpathfilter**, **outroutemap** and **sendcommunity** take effect immediately, for all other parameters in this command to take effect, you must already have enabled automatic updating using the **enable bgp autosoftupdate** command. Otherwise, you will need to trigger an update, using the **reset bgp peer soft** command, in order for the changes to take effect.

Parameter	Description	
PEer	The IP address of the peer, in dotted decimal notation. Default: no default	
AUthentication	Whether to use MD5 authentication for the BGP peer. If you specify **md5**, you must also specify **password**. Default: **none**	
	MD5	An MD5 digest is added to every BGP packet sent over the TCP connection and is authenticated at the other end. If any part of the digest cannot be verified, the packet is dropped with no response sent.
	NONE	The BGP session is not authenticated.
CAPAbilitymatching	Whether the local BGP speaker (the router) strictly or loosely compares this remote peer's capabilities with its own capabilities. Default: **loose**	
	STrict	The local speaker only establishes a session to the peer if the local speaker supports all the capabilities that the remote peer advertises in its Open message.
	LOOSE	The local speaker and the peer establish a session as long as the remote peer supports at least the IPv4 address family and IPv4 unicast capabilities.
CLIEnt	Whether the peer is a client of the router if the router is a route reflector (RR). RRs selectively advertise routes they learn from their IBGP peers to their other IBGP peers. The router is a route reflector if it has at least one client peer, meaning that **client** is **yes** for at least one of its peers. For more information about route reflection, and client and non-client peers, see "How to Improve IBGP Scalability" on page 27-33. Route reflection is only valid for IBGP peers, so **client=yes** is only valid when the local ASN and the remote ASN are the same. Default: **no**	
	NO	The peer is a non-client peer of the RR. When the RR receives a route from a non-client peer, the RR reflects the route only to its client peers, not to any other non-client peers it has.
	YES	The peer is a client peer of the RR. When the RR receives a route from a client peer, the RR reflects the route to all its peers, both client and non-client.
CONnectretry	The time interval between attempts to establish a BGP connection to the peer, in seconds. Default: **120**	
	0	The router does not repeat an attempt to establish a BGP connection.
	1..4294967295	The router waits the specified number of seconds between attempts.
	DEFault	The router waits 120 seconds between attempts.

Parameter (cont)	**Description** (cont)	
DEFaultoriginate	Whether to advertise the default route (0.0.0.0/0) to this peer, when the router's BGP routing table contains the default route. To advertise the default route, you need to do all of the following: • set this parameter to **yes** • create the default route on the router (or the router needs to learn it from another routing source) • configure BGP with an import or network entry that includes the default route • import the default route into the BGP routing table by using the **enable bgp defaultoriginate** command Default: **no** (the default route is not propagated from the router's BGP routing table to the peer's RIB)	
DESCription	A description of the peer, which has no effect on its operation. A string from 1 to 63 characters long. All printable characters are valid except the question mark and double quotes. If *description* contains spaces, the string must be in double quotes. Default: **none**	
EHOps	The number of hops put in the *TTL* (Time To Live) field of BGP messages for external BGP. Normally, EBGP requires that BGP peers be connected to a common network, which means they are separated by a single hop. Setting **ehops** to a value greater than 1 indicates that multihop EBGP is allowed. Default: **1**	
	1..255	The specified number of hops is put into the TTL field.
	DEFault	The number of hops put in the TTL field is 1.
FASTfallover	Whether fast fallover is enabled on the link to the peer. If fast fallover is enabled, the peer session is reset as soon as the interface that supports the session goes down. If fast fallover is disabled, the session is reset only when its keepalive timer expires. Default: **no** (fast fallover is disabled)	
HOLdtime	The value in seconds that this router proposes for the time interval between reception of keepalive and/or update messages from this peer. The actual hold time used on a peer connection is negotiated when the connection is opened, as the lower of the hold times proposed. Default: **90**	
	0	This router proposes not to have a hold time on this BGP connection.
	3..65535	This router proposes the specified number of seconds as hold time.
	DEFault	This router proposes a hold time of 90 seconds.

Parameter (cont)	**Description** (cont)
INFilter	The prefix list that filters update messages that the router receives from this peer. If a prefix matches a prefix in the prefix list, BGP rejects that route. Otherwise, it accepts the route. The prefix list must already exist. To create a prefix list, use the **add ip prefixlist** command on page 28-45 of Chapter 28, Filtering IP Routes. If you specify more than one of **inpathfilter**, **infilter** and **inroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the prefix, so a prefix is only included if all the applied filters result in it being included. Default: **none**
INPathfilter	The AS path list that filters the BGP update messages from this peer. You can use an AS path list to exclude or include update messages that have traversed particular ASs or paths. If the path list does not already exist, it is created. To create a path list and/or add entries to it, use the **add ip aspathlist** command on page 28-41 of Chapter 28, Filtering IP Routes. If you specify more than one of **inpathfilter**, **infilter** and **inroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**
INRoutemap	The route map that filters and/or modifies prefixes from this peer. You can use a route map to include or exclude update messages or a subset of an update message's routes, on the basis of a range of BGP attributes, and/or to modify attributes. If you specify the inroutemap parameter without specifying a route map name, the current route map is disassociated from the peer. The route map must already exist. To create a route map use the **add ip routemap** command on page 28-50 of Chapter 28, Filtering IP Routes. If you specify more than one of **inpathfilter**, **infilter** and **inroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**

Parameter (cont)	**Description** (cont)	
KEEpalive	The time in seconds that this router would prefer to leave between keepalive messages to this peer. This time should be one third of the **holdtime** parameter. The actual value used for the keep alive interval is determined once the BGP connection is opened, because the hold time interval is calculated as part of the BGP connection opening. The actual keep alive interval is calculated so that the ratio: configured keep alive interval: configured hold time interval is the same as the ratio: actual keep alive interval: negotiated hold time interval. If the hold time is negotiated at 0 seconds, then the keep alive interval is also 0 seconds, and keepalive messages are not sent. Default: one third of **holdtime**	
	1..21845	This router prefers the specified number of seconds as keepalive interval.
	DEFault	This router prefers a keepalive interval of one third the hold time.
LOCal	The local interface. In certain circumstances, the router uses this address as the source for BGP packets it generates and sends to this BGP peer. For a description of when the router uses the local interface, see "How to Set the IP Address that Identifies the Router" on page 27-40. Default: **none**	
MAXPREFIX	The maximum number of network prefixes that the router expects to receive from this peer. This parameter provides a safety mechanism in case the peer sends more prefixes than you might normally expect to receive. Default: **off**	
	1..4294967295	The maximum number of prefixes the router expects to receive from this peer. Once this number is exceeded, the action you specify in **maxprefixaction** is carried out.
	OFF	No maximum prefix checking.
MAXPREFIXAction	The action to take when a peer has sent a number of prefixes that exceeds the number specified by **maxprefix**. Default: **warning**	
	Warning	The router logs warnings when the maximum number of prefixes is exceeded.
	Terminate	The router resets the peer connections and logs warnings.
MINAsoriginated	The minimum time in seconds between advertisements, from the router to this peer, of routes that originate in the router's autonomous system. Default: **15**	
	0..3600	The interval is the specified number of seconds.
	DEFault	The interval is 15 seconds.

Parameter (cont)	**Description** (cont)	
MINRouteadvert	The minimum time in seconds between advertisements, from the router to this peer, of routes that originate outside the router's autonomous system. Default: **30**	
	0..3600	The interval is the specified number of seconds.
	DEFault	The interval is 30 seconds.
NEXthopself	Whether this router advertises to this peer that the next hop for all routes is itself. Default: **no**	
	YES	All updates that the router sends to this peer specify this router as the next hop.
	NO	The next hop is specified as described in RFC 1771.
OUTFilter	The prefix list that filters update messages that the router sends to the peer. If a prefix matches a prefix in the prefix list, BGP removes that route from the update message. Otherwise, it leaves the route in the update message and therefore advertises it to the peer. The prefix list must already exist. To create a prefix list, use the **add ip prefixlist** command on page 28-45 of Chapter 28, Filtering IP Routes. If you specify more than one of **outpathfilter**, **outfilter** and **outroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the prefix, so a prefix is only included if all the applied filters result in it being included. Default: **none**	
OUTPathfilter	The AS path list that filters the BGP update messages sent to this peer. You can use an AS path list to exclude or include update messages that have traversed particular ASs or paths. If the path list does not already exist, it is created. To create a path list and/or add entries to it, use the **add ip aspathlist** command on page 28-41 of Chapter 28, Filtering IP Routes. If you specify more than one of **outpathfilter**, **outfilter** and **outroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**	

Parameter (cont)	**Description** (cont)
OUTRoutemap	The route map that filters and/or modifies prefixes sent to this peer. You can use a route map to include or exclude update messages or a subset of an update message's routes, on the basis of a range of BGP attributes, and/or to modify attributes. The route map must already exist. To create a route map use the **add ip routemap** command on page 28-50 of Chapter 28, Filtering IP Routes. If you specify the **outroutemap** parameter without specifying a route map name, the current route map is disassociated from the peer. If you specify more than one of **outpathfilter**, **outfilter** and **outroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**
PASSword	The key used by the authentication algorithm. Two BGP peers can communicate with each other only if they have the same key. *password* is a character string from 1 to 80 characters long. All printable characters are valid except the question mark and double quotes. If *password* contains spaces, it must be in double quotes. Only valid if **authentication=md5** Default: no default
POLICYTemplate	The ID number of the peer policy template that applies to this peer. The specified policy template must already exist. To create a template, use the **add bgp peertemplate** command. You can only specify **remoteas**, **description**, **authentication**, **password**, **fastfallover,** and **ehops** at the same time as **policytemplate**. The template provides all other configuration values. Specifying **policytemplate=** with no number disassociates the template and the peer. The peer retains the template's settings. You can then modify the peer.
PRIVateasfilter	Whether private AS numbers (from 64512 to 65535) are stripped from the AS PATH attribute on update messages the router sends to the peer. Default: **no**
REMoteas	The remote Autonomous System to which this peer belongs. If the remote AS number is the same as this router's AS number, the peer is an internal BGP (IBGP) peer. If the remote AS number is different from this router's AS number, the peer is an external BGP (EBGP) peer. If the remote AS numbers are different but the routers have the same confederation peer, the peer is a confederation BGP peer. The AS number is assigned by the IANA. Default: no default

Parameter (cont)	**Description** (cont)	
SENdcommunity	Whether the router includes the community attribute in update messages that it sends to this peer. Default: **no**	
	YES	The community attribute is set in update messages to this peer. To set the value of the community attribute, create a route map with a **set** clause to set the community, and use the **outroutemap** parameter to apply it to update messages to this peer. To create a route map use the **add ip routemap** command on page 28-50 of Chapter 28, Filtering IP Routes.
	NO	The community attribute is not set in update messages to this peer, even if it is set in the route map used by the peer.

Examples

To set authentication for a BGP peer whose IP address is 192.168.1.1, use the command:

```
set bgp pe=192.168.1.1 au=md5 password=a-very-secret-password
```

To modify the keepalive time and hold time for a BGP peer whose IP address is 192.168.1.1, use the command:

```
set bgp pe=192.168.1.1 kee=10 hol=30
```

Related Commands

add bgp peer
add ip aspathlist in Chapter 28, Filtering IP Routes
add ip filter in Chapter 22, Internet Protocol (IP)
add ip routemap in Chapter 28, Filtering IP Routes
delete bgp peer
disable bgp peer
enable bgp autosoftupdate
enable bgp peer
reset bgp peer
reset bgp peer soft
show bgp peer

set bgp peertemplate

Syntax

```
SET BGP PEERTemplate=1..30 [CLIEnt={NO|YES}]
   [CONnectretry={DEFault|0..4294967295}]
   [DESCription={NONE|description}]
   [HOLdtime={DEFault|0|3..65535}]
   [INFilter={NONE|prefixlist-name}]
   [INPathfilter={NONE|1..99}]
   [INRoutemap={NONE|routemap}]
   [KEEpalive={DEFault|1..21845}] [LOCal={NONE|1..15}]
   [MAXPREFIX={OFF|1..4294967295}]
   [MAXPREFIXAction={Terminate|Warning}]
   [MINAsoriginated={DEFault|0..3600}]
   [MINRouteadvert={DEFault|0..3600}]
   [NEXthopself={NO|YES}]
   [OUTFilter={NONE|prefixlist-name}]
   [OUTPathfilter={NONE|1..99}]
   [OUTRoutemap={NONE|routemap}]
   [PRIVateasfilter={NO|YES}] [SENdcommunity={NO|YES}]
```

Description This command modifies a template for use on BGP peers. BGP applies the changes to all peers that are using the template.

Parameter	Description	
PEERTemplate	The ID number of the template. Default: no default	
CLIEnt	Whether peers that use the template are clients of the router if the router is a route reflector (RR). RRs selectively advertise routes they learn from their IBGP peers to their other IBGP peers. The router is a route reflector if it has at least one client peer, meaning that **client** is **yes** for at least one of its peers. See "How to Improve IBGP Scalability" on page 27-33 for more information about route reflection, and client and non-client peers. Route reflection is only valid for IBGP peers, so **client=yes** is only valid when the local ASN and the remote ASN are the same. Default: **no**	
	NO	Peers that use the template are non-client peers of the RR. When the RR receives a route from a non-client peer, the RR reflects the route only to its client peers, not to any other non-client peers it has.
	YES	Peers that use the template are client peers of the RR. When the RR receives a route from a client peer, the RR reflects the route to all its peers, both client and non-client.
CONnectretry	The time interval between attempts to establish a BGP connection to peers that use the template, in seconds. Default: **120**	
	0	The router does not repeat an attempt to establish a BGP connection.
	1..4294967295	The router waits the specified number of seconds between attempts.
	DEFault	The router waits 120 seconds between attempts.

Parameter (cont)	**Description** (cont)
DESCription	A description for the peers that use the template, which has no effect on their operation. A string 1 to 63 characters long. All printable characters are valid except the question mark and double quotes. If *description* contains spaces, the string must be in double quotes. Default: **none**
HOLdtime	The value in seconds that this router proposes for the time interval between reception of keepalive and/or update messages from peers that use the template. The actual hold time used on a peer connection is negotiated when the connection is opened, as the lower of the hold times proposed. Default: **90**
0	This router proposes not to have a hold time on this BGP connection.
3..65535	This router proposes the specified number of seconds as hold time.
DEFault	This router proposes a hold time of 90 seconds.
INFilter	The prefix list that filters update messages that the router receives from peers that use the template. If a prefix matches a prefix in the prefix list, BGP rejects that route. Otherwise, it accepts the route. The prefix list must already exist. To create a prefix list, use the **add ip prefixlist** command on page 28-45 of Chapter 28, Filtering IP Routes. If you specify more than one of **inpathfilter**, **infilter** and **inroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the prefix, so a prefix is only included if all the applied filters result in it being included. Default: **none**
INPathfilter	The AS path list that filters the BGP update messages from peers that use the template. You can use an AS path list to exclude or include update messages that have traversed particular ASs or paths. If the path list does not already exist, it is created. To create a path list and/or add entries to it, use the **add ip aspathlist** command on page 28-41 of Chapter 28, Filtering IP Routes. If you specify more than one of **inpathfilter**, **infilter** and **inroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**
INRoutemap	The route map that filters and/or modifies prefixes from peers that use the template. You can use a route map to include or exclude update messages or a subset of an update message's routes, on the basis of a range of BGP attributes, and/or to modify attributes. The route map must already exist. To create a route map use the **add ip routemap** command on page 28-50 of Chapter 28, Filtering IP Routes. If you specify more than one of **inpathfilter**, **infilter**, and **inroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**

Parameter (cont)	**Description** (cont)	
KEEpalive	The time in seconds that this router would prefer to leave between keepalive messages to peers that use the template. This time should be one third of the **holdtime** parameter. The actual value used for the keep alive interval is determined once the BGP connection is opened, because the hold time interval is calculated as part of the BGP connection opening. The actual keep alive interval is calculated so that the ratio: configured keep alive interval: configured hold time interval is the same as the ratio: actual keep alive interval: negotiated hold time interval. If the hold time is negotiated at 0 seconds, then the keep alive interval is also 0 seconds, and keepalive messages are not sent. Default: one third of **holdtime**	
	1..21845	This router prefers the specified number of seconds as keepalive interval.
	DEFault	This router prefers a keepalive interval of one third the hold time.
LOCal	The local interface. In certain circumstances, the router uses this address as the source for BGP packets it generates and sends to peers that use this template. For a description of when the router uses the local interface, see "How to Set the IP Address that Identifies the Router" on page 27-40. Default: **none**	
MAXPREFIX	The maximum number of network prefixes that the router expects to receive from peers that use the template. This parameter provides a safety mechanism in case the peer sends more prefixes than you might normally expect to receive. Default: **off**	
	1..4294967295	The maximum number of prefixes the router expects to receive from peers that use the template. Once this number is exceeded, the action you specify in **maxprefixaction** is carried out.
	OFF	No maximum prefix checking.
MAXPREFIXAction	The action to take when a peer has sent a number of prefixes that exceeds the number specified by **maxprefix**. Default: **warning**	
	Warning	The router logs warnings when the maximum number of prefixes is exceeded.
	Terminate	The router resets the peer connections and logs warnings.
MINAsoriginated	The minimum time in seconds between advertisements, from the router to peers that use the template, of routes that originate in the router's autonomous system. Default: **15**	
	0..3600	The interval is the specified number of seconds.
	DEFault	The interval is 15 seconds.

Parameter (cont)	**Description** (cont)	
MINRouteadvert	The minimum time in seconds between advertisements, from the router to peers that use the template, of routes that originate outside the router's autonomous system. Default: **30**	
	0..3600	The interval is the specified number of seconds.
	DEFault	The interval is 30 seconds.
NEXthopself	Whether this router advertises to peers that use the template that the next hop for all routes is itself. Default: **no**	
	YES	All updates that the router sends to peers that use this template specify this router as the next hop.
	NO	The next hop is specified as described in RFC 1771.
OUTFilter	The prefix list that filters update messages that the router sends to peers that use this template. If a prefix matches a prefix in the prefix list, BGP removes that route from the update message. Otherwise, it leaves the route in the update message and therefore advertises it to the peer. The prefix list must already exist. To create a prefix list, use the **add ip prefixlist** command on page 28-45 of Chapter 28, Filtering IP Routes. If you specify more than one of **outpathfilter**, **outfilter** and **outroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the prefix, so a prefix is only included if all the applied filters result in it being included. Default: **none**	
OUTPathfilter	The AS path list that filters the BGP update messages sent to peers that use this template. You can use an AS path list to exclude or include update messages that have traversed particular ASs or paths. If the path list does not already exist, it is created. To create a path list and/or add entries to it, use the **add ip aspathlist** command on page 28-41 of Chapter 28, Filtering IP Routes. If you specify more than one of **outpathfilter**, **outfilter** and **outroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**	

Parameter (cont)	**Description** (cont)	
OUTRoutemap	The route map that filters and/or modifies prefixes sent to peers that use this template. You can use a route map to include or exclude update messages or a subset of an update message's routes, on the basis of a range of BGP attributes, and/or to modify attributes. The route map must already exist. To create a route map use the **add ip routemap** command on page 28-50 of Chapter 28, Filtering IP Routes. If you specify more than one of **outpathfilter**, **outfilter** and **outroutemap**, the router applies them in that order: first the AS path filter, then the prefix filter, then the route map. Note that the router stops checking after the first filter entry that excludes the update, so an update is only included if all the applied filters result in it being included. Default: **none**	
PRIVateasfilter	Whether private AS numbers (from 64512 to 65535) are stripped from the AS PATH attribute on update messages the router sends to peers that use this template. Default: **no**	
SENdcommunity	Whether the router includes the community attribute in update messages that it sends to peers that use this template. Default: no	
	YES	The community attribute is set in update messages to peers that use this template. To set the value of the community attribute, create a route map with a **set** clause to set the community, and use the **outroutemap** parameter to apply it to update messages to peers that use this template. To create a route map use the **add ip routemap** command on page 28-50 of Chapter 28, Filtering IP Routes.
	NO	The community attribute is not set in update messages to this peer, even if it is set in the route map used by the peers that use this template.

Examples

To modify a peer policy template 1 to have a hold time of 30 seconds, use the command:

```
set bgp peert=1 hol=30
```

Related Commands

add bgp peer
add bgp peertemplate
set bgp peer
show bgp peer
show bgp peertemplate

set ip autonomous

Syntax `SET IP AUtonomous=1..65534`

Description This command sets the router's autonomous system number (ASN). The router cannot be configured to use BGP-4 until it is part of an AS and therefore has an ASN.

There are two types of ASNs:

- public ASNs (1 to 64511)

  These are globally unique and assigned by the IANA. They identify the router's AS when the router exchanges routes with external organisations.

- private ASNs (64512 to 65534)

  These are non-assigned numbers. You can use them when you are running BGP in an AS confederation, for example. The individual AS numbers in the confederation can be non-assigned numbers.

Caution If the router has a peer relationship with a public peer, always use an assigned autonomous system number rather than inventing one.

Related Commands

add bgp peer
enable bgp peer
show bgp

show bgp

Syntax `SHow BGP`

Description This command displays information about BGP global configuration and operation (Figure 27-20, Table 27-12).

Figure 27-20: Example output from the **show bgp** command

```
BGP router ID ................. 192.168.1.1
BGP Cluster ID ................ 192.168.1.1
Local autonomous system ....... 123
Confederation ID .............. 1234
Local preference .............. 100 (default)
Multi Exit Discriminator ...... -
Route table route map ......... -
Auto soft reconfiguration ..... Disabled
Default route origination ..... Disabled
Auto summary .................. Disabled

Number of peers
  Defined ..................... 4
  Established ................. 2
BGP route table
  Iteration ................... 231
  Number of routes ............ 12654
  Route table memory .......... 431872

BGP route flap damping ........ Enabled
```

Table 27-12: Parameters in output of the **show bgp** command

Parameter	Meaning
BGP router ID	The ID for BGP for this router. This is the router ID if one has been set using the **set bgp** command. Otherwise it is the local interface if one has been configured. Otherwise it is the highest IP address configured on any of the router's interfaces. For more information, see "How to Set the IP Address that Identifies the Router" on page 27-40.
BGP Cluster ID	The cluster ID of the cluster for which the router is a route reflector (RR). See "How to Improve IBGP Scalability" on page 27-33 for more information.
Local autonomous system	The number of the Autonomous System to which this router belongs.
Confederation ID	The AS number of the AS confederation to which this router belongs.
Local preference	The value of the local preference for this router. This is sent to internal BGP peers to help in the decision process for deciding which BGP routes go into the main routing table.
Multi-Exit Discriminator	The value of the multi-exit discriminator for this router. This is sent to external BGP peers to help in the decision process for deciding which BGP routes go into the main routing table.
Route table route map	The name of the route map that BGP uses before entering a route into the route table.

Table 27-12: Parameters in output of the **show bgp** command (cont)

Parameter	Meaning
Auto soft reconfiguration	Whether the router automatically updates modified peers.
Default route origination	Whether BGP imports the default route (0.0.0.0/0) into its routing table when both of the following conditions occur: • the default route is present in the router's RIB • BGP is configured with an import or network entry that includes the default route.
Auto summary	Whether the router automatically summarises locally originated or imported subnet routes into a classful network route; one of Enabled or Disabled.
Number of peers	Counters giving the number of configured and established peers.
Defined	The number of peers currently configured on the router.
Established	The number of peers currently in the established state.
BGP route table	Information about the BGP route table.
Iteration	The number of times the BGP route table has been modified.
Number of routes	The number of routes in the BGP route table.
Route table memory	The amount of router memory currently used in the BGP route table.
BGP route flap damping	Whether route flap damping is enabled.

Examples

To show general BGP parameters and a summary of BGP operations, use the command:

```
sh bgp
```

Related Commands

enable bgp autosoftupdate
enable bgp autosummary
enable bgp defaultoriginate
show bgp aggregate
show bgp import
show bgp network
show bgp peer
show bgp route

show bgp aggregate

Syntax `SHow BGP AGGregate`

Description This command displays information about the BGP aggregate entries configured in this router (Figure 27-21, Table 27-13).

Figure 27-21: Example output from the **show bgp aggregate** command

```
BGP aggregate entries

Prefix               Summary   Route map
-----------------------------------------------
192.168.248.0/21     Yes       aggregate_map
192.168.16.0/21      No        -
-----------------------------------------------
```

Table 27-13: Parameters in output of the **show bgp aggregate** command

Parameter	Meaning
Prefix	Prefix for this aggregate entry. This is the prefix that BGP advertises for this aggregate as long as a route that is a subset of this prefix is present in the BGP routing table.
Summary	Either Yes or No. If yes, the aggregate route is advertised and no subset routes. If no, subset routes are also advertised.
Route map	Name of the route map used to filter routes for this aggregate entry and to set the BGP attributes for the aggregate route entry.

Examples To display information about BGP aggregates, use the command:

```
sh bgp agg
```

Related Commands

add bgp aggregate
delete bgp aggregate
set bgp aggregate
show bgp
show ip routemap in Chapter 28, Filtering IP Routes

show bgp confederation

Syntax `SHow BGP CONfederation`

Description This command displays information about the BGP confederation setup of this router (Figure 27-22, Table 27-14).

Figure 27-22: Example output from the **show bgp confederation** command

```
BGP confederation information

Local AS ................ 60001
Confederation ID ........ 1234
Confederation peers ..... 60002
                          60003
Peers ................... 192.168.1.1 (AS 60001, IBGP)
                          192.169.3.2 (AS 60002, CBGP)
                          192.170.4.5 (AS 7658, EBGP)
```

Table 27-14: Parameters in output of the **show bgp confederation** command

Parameter	Meaning	
Local AS	The AS number of the AS to which this router belongs.	
Confederation ID	The AS number of the AS confederation to which this router belongs. The AS confederation behaves as this AS to external BGP peers.	
Confederation peers	The AS numbers of Autonomous Systems in the router's AS confederation.	
Peers	A list of the configured BGP peers from the point of view of AS confederation configuration. For each peer, the peer address, peer AS, and BGP type is given. BGP type is:	
	EBGP	External BGP. The peer is in a different AS and outside the AS confederation.
	CBGP	Confederation BGP. The peer is in a different AS but inside the AS confederation.
	IBGP	Internal BGP. The peer is in the same AS as this router.

Examples To display information concerning the AS confederation to which this router belongs, use the command:

```
sh bgp con
```

Related Commands

add bgp confederationpeer
delete bgp confederationpeer
set bgp
set ip autonomous
show bgp

show bgp backoff

Syntax `SHow BGP BACkoff`

Description This command displays BGP backoff details (Figure 27-23, Table 27-15).

Figure 27-23: Example output of the **show bgp backoff** command

```
BGP Backoff Stats:
 Stat                          Value
---------------------------------
command status              ENABLED
backOff state                NORMAL
total hist backOffs               5
total backOffs                    0
total backOff Limit               0
consecutive backOffs              0
consecutive backOffs limit        5
base Timeout                     10
Timeout multiplier              100%
Timeout step                      1
Timeout length (sec)             10
Mem Upper Threshold Value        95%
Mem Upper Notify               TRUE
Mem Lower Threshold Value        90%
Mem Lower Notify              FALSE
Current Mem use                  84%
---------------------------------
```

Table 27-15: Parameters in output of the **show bgp backoff** command

Parameter	Meaning
command status	Whether BGP backoff is enabled.
backOff state	The current status of BGP backoff. • NORMAL displays when BGP backoff is not active and BGP is processing normally. • BACKED OFF displays when system memory use has reached its upper threshold and BGP processing is halted. • PEER DISABLED displays when the consecutive or total backoff limits have been reached and the peers have been disabled.This also displays if BGP backoff is enabled, but no peer has yet been discovered. • DISABLED displays when the user has disabled backoff functionality.
total hist backOffs	The total number of backoffs that have occurred. Unlike "total backOffs", this value is not reset when BGP disables peers because it reaches the total or consecutive backoff limit. You can use this value to determine the optimal setting for the total backoff limit.
total backOffs	The number of times that BGP has backed off since it last reached the total backoff limit. Note that this counter is not reset when BGP disables its peers because the consecutive backoff limit is reached.

Table 27-15: Parameters in output of the **show bgp backoff** command (cont)

Parameter	Meaning
total backOff limit	The total number of backoffs that cause BGP to disconnect its peers.
consecutive backOffs	The number of times in a row that BGP has reached the end of its backoff time and found that system memory is high enough that it backs off again immediately.
consecutive backOffs limit	The number of consecutive backoffs that causes BGP to disable all peers.
base Timeout	The number of seconds used to calculate the total backoff time for the first backoff iteration. The first backoff time is calculated as: `basetime x multiplier/100`
Timeout multiplier	A multiplier for increasing or decreasing the backoff time at each backoff iteration. The change in backoff time at each step is calculated as: `current backoff time x multiplier/100`
Timeout step	The number of backoff iterations after which the backoff time is recalculated.
Timeout length	The current backoff time.
Mem Upper Threshold Value	The percentage of system memory use that triggers BGP to back off. This threshold is set with the **backoff** parameter.
Mem Upper Notify	Whether BGP is monitoring the upper or lower thresholds of the system memory use. When TRUE, BGP is monitoring the upper threshold and its state is NORMAL.
Mem Lower Threshold Value	The percentage of system memory use that the router must fall below before BGP backoff will end. This threshold is set using the **low** parameter.
Mem Lower Notify	Whether BGP is monitoring the upper or lower threshold of the system memory use. When TRUE, BGP is monitoring the lower threshold and is in a BACKED OFF or PEER DISABLED state.
Current Mem use	The amount of memory used by the system at the moment the command was executed.

Example To see the existing BGP backoff settings, use the command:

```
sh bgp bac
```

Related Commands

disable bgp backoff
enable bgp backoff
set bgp backoff
show bgp memlimit

show bgp counters

Syntax `SHow BGP COUnters[={RIB|UPdate|DB|DB-All|PROCess|NEXThop}[,...]]`

Description This command displays counter information for BGP.

Figure 27-24: Example output of the **show bgp counters=update** command

```
Update Counters:
---------------------------------------------------

  Update Message:
  Header too small ................ 0
  Header too long ................. 0
  Withdrawn too long .............. 0
  Total Path too long ............. 0

  Prefix:
  NLRI error ...................... 0
  Withdrawn errors ................ 0
    Mask > 32bits ................. 0
    Data too long ................. 0
    Invalid Address ............... 0

  Path attributes ................. 0
    Data shorter .................. 0
    Seen twice .................... 0
    Missing mandatory ............. 0

    Origin ........................ 0
      Length wrong ................ 0
      Flags wrong ................. 0
      Unknown origin .............. 0

    AS path ....................... 0
    Silently dropped .............. 0
      Flags wrong ................. 0
      List get failed ............. 0
      Unknown Seg type ............ 0
      Non-confed peer ............. 0
      Data too long ............... 0
      Data too short .............. 0
      AS path loop ................ 0
      Confed seg order ............ 0

    Next hop ...................... 0
      Length wrong ................ 0
      Flags wrong ................. 0
      Address Zero ................ 0
      Interface found ............. 0
```

Figure 27-24: Example output of the **show bgp counters=update** command (cont)

```
    Med ............................ 0
      Length wrong ................. 0
      Flags wrong .................. 0

    Local preference ............... 0
      Length wrong ................. 0
      Flags wrong .................. 0
      External peer ................ 0

    Atomic aggregate ............... 0
      Length wrong ................. 0
      Flags wrong .................. 0

    Aggregate ...................... 0
      Length wrong ................. 0
      Flags wrong .................. 0

    Community ...................... 0
      Length wrong ................. 0
      Flags wrong .................. 0

    Originator ..................... 0
      Length wrong ................. 0
      Flags wrong .................. 0
      From eBGP Peer ............... 0
      Loops detected ............... 0

    Cluster List ................... 0
      Flags wrong .................. 0
      From eBGP Peer ............... 0
      Loops detected ............... 0

    Unknown Attributes ............. 0
      Flag wrong ................... 0
      Non-transitive ............... 0
      Transitive ................... 0

  Memory:
    Low memory drops ............... 0

  Filter:
    Path exclude ............... 67022
    Prefix exclude ................. 0
    Routemap exclude ............... 0

  Route Selection Fail ............. 0
    Match List empty ............... 0
    Select List empty .............. 0
    NextHop No Route ............... 0

  Internal Control:
    Control Pointers ............... 0
    Message Pointer ................ 0
    Dropped Pointer ................ 0
```

Figure 27-25: Example output of the **show bgp counters=rib** command

```
Total Nodes: 290268
Split Nodes: 135002
Paths:       155266
Withdrawn Paths: 0
Aggregate Paths: 0
nexthop List size: 1
interface route List size: 0
routemap cache List size: 0
BGP Idle Flag: 00000000
Free buffers: 57193

RIB Counters:
--------------------------------------------------

Add to IPG:
  Peer lookup failed ..............  0
  Next hop find failed ............  0
  Next hop no route ...............  0
  IP Log Index NULL ...............  0
  Add failed ......................  0
Delete all from peer:
  Walk pointer NULL ...............  0

Interface list:
  Delete search failed ............  0

Next hop list:
  Delete search failed ............  0

Node Copy:
  Sending route NULL ..............  0
  IP route NULL ...................  0

Withdrawn Route:
  Unlink Error ....................  0
```

Figure 27-26: Example output of the **show bgp counters=db** command

```
AsPathSegDb Entries: 25741
AsSegListDb Entries: 25741
PathAttribDb Entries: 25741
PrefixDb Entries: 0
UnknownAttribDb Entries: 0
UnknownAttribListDb Entries: 0
CommunityListDb Entries: 0
```

Figure 27-27: Example output of the **show bgp counters=db-all** command

```
AsPathSegDb Entries: 25741
AsSegListDb Entries: 25741
PathAttribDb Entries: 25741
PrefixDb Entries: 0
UnknownAttribDb Entries: 0
UnknownAttribListDb Entries: 0
CommunityListDb Entries: 0
```

Figure 27-28: Example output of the **show bgp counters=process** command

```
Last run process: SEND_REACHABLES
Current running process: none
Current waiting processes:

Process back-off metric: 1

Process Stats:
Process                      Start   Continue       Time
-----------------------------------------------------------
DELETE_ALL_FROM_PEER             9     943852          4
IMPORTS_RECHECK                  0          0          0
NEXT_HOPS_RECHECK            16503          0          0
ROUTE_SELECTION              16626   40129126        490
RIB_IN_WITHDRAWN            132933   19489261        287
RIB_IN_REACHABLES          9178505   22910398       3532
UDPATE_MESSAGE             9311509          0       1013
UDPATE_IPG_TABLES          1307362          0        304
SEND_WITHDRAWN              429911          0          5
FLAG_REACHABLES            1304726          0         10
FLAG_UNSYNCED_PEERS         777554          0          3
SEND_REACHABLES          100858683          0       2580
```

Figure 27-29: Example output of the **show bgp counters=nexthop** command

```
nexthop List size: 1
Next hop: 211.30.1.1
  have ip route: 1
    ip route: 0.0.0.0
    ip int index:      1 log int index: 0
    ip metric: 1
  routePt: 06aac0f8
  currentState: 1
  ipRouteRefCount: 167450016
```

show bgp damping

Syntax `SHow BGP DAMping`

Description This command displays information about the BGP route flap damping configuration and operation (Figure 27-30, Table 27-16).

Figure 27-30: Example output from the **show bgp damping** command

```
BGP Route Flap Damping
  Status ..................... ENABLED
  Routes in Engine ........... 40
    Monitored Routes ......... 2
    Suppressed Routes ........ 9
  Forgotten Routes ........... 46

Parameterset 0
  Default configuration
  Current State ..... DISABLED
  Suppression ....... 2000        Reuse .......... 750
  Half life ......... 15 min      Maximum Hold ... 1:4

Parameterset 1
  Severely penalise unreachable to test network
  Current State ..... DISABLED
  Suppression ....... 1200        Reuse .......... 500
  Half life ......... 15 min      Maximum Hold ... 1:4

Parameterset 6
  <Parameterset6>
  Current State ..... ENABLED
  Suppression ....... 1500        Reuse .......... 950
  Half life ......... 10 min      Maximum Hold ... 1:5
```

Table 27-16: Parameters in output of the **show bgp damping** command

Parameter	Meaning
Status	Whether BGP route flap damping is enabled on the router.
Monitored Routes	Number of routes that are not suppressed but are being monitored by the suppression engine.
Routes in Engine	Number of routes for which the suppression engine is currently maintaining an FoM.
Suppressed Routes	Number of routes that are currently being suppressed by the suppression engine.
Forgotten Routes	Number of routes that incurred a damping penalty in the past, but have had that penalty forgotten due to those routes experiencing a sufficient period of stability. Note that if the same route is forgotten more than once, it is counted more than once. The counter does not decrement when a previously forgotten route incurs a new damping penalty.
Parameterset n	ID number of the parameter set.
[Description line]	The user-defined description of the parameter set. If the parameter set has no description, the display is "<Parameterset*n*>" where *n* is the number of the parameter set.

Table 27-16: Parameters in output of the **show bgp damping** command (cont)

Parameter	Meaning
Current state	Whether the parameter set is enabled.
Suppression	FoM value above which a route advertisement is suppressed.
Reuse	FoM value below which a suppressed route becomes selectable again.
Half Life	Time interval within which the route's FoM will halve if the route remains stable, in minutes.
Maximum Hold	Ratio of half life to the maximum time a route may be suppressed for, regardless of its stability history. For example, if Half Life is 15 and Maximum Hold is 1:4, a route becomes available again after 60 minutes, even if its FoM still exceeds the Reuse value.

Examples

To check if parameter set 3 is enabled, use the command:

```
sh bgp dam
```

Related Commands

create bgp damping parameterset
disable bgp damping
enable bgp damping
show bgp damping routes

show bgp damping routes

Syntax `SHow BGP DAMping ROUtes`

Description This command displays information about the routes in the route flap damping suppression engine. It lists all monitored and suppressed routes (Figure 27-31, Table 27-17).

Figure 27-31: Example output from the **show bgp damping routes** command

```
-------------------------------------------------------------------------------
Par  Prefix/Mask         Next Hop     Current     Pen    Num   Last St   Next St
Set                                     State   (FoM)  Flaps    Change    Change
-------------------------------------------------------------------------------
  0  192.168.5.0/24      1.1.1.1        >eM     992      1  00:00:10  01:23:40
  0  192.168.10.0/24     1.1.1.1        >eM     992      1  00:00:10  01:23:40
  0  192.168.7.0/24      1.1.1.1        >eS    2961      3  00:00:20  00:29:45
  0  192.168.3.0/24      1.1.1.1        >eS    4938      5  00:00:20  00:40:50
  0  192.168.9.0/24      1.1.1.1        >eM     992      1  00:00:10  01:23:40
  0  192.168.6.0/24      1.1.1.1        >eM    1976      2  00:00:20  01:38:40
  0  192.168.4.0/24      1.1.1.1        >eS    1984      2  00:00:10  00:21:05
-------------------------------------------------------------------------------
```

Table 27-17: Parameters in output of the **show bgp damping routes** command

Parameter	Meaning
Par Set	ID of the parameter set used to maintain the FoM for the given route.
Prefix/Mask	Network IP address and CIDR mask of the given route.
Next Hop	IP address of the next hop for the given route.
Current State	Current state of the given route: • Status flags: > the best route for the given prefix * next hop unreachable a aggregate route s aggregate suppressed • Origin flags: i internal e external ? incomplete ! unreachable or withdrawn • Damping flags: S Damping suppressed M Damping monitored
Pen (FoM)	Current FoM value for the route.
Num Flaps	Number of times the route has become unreachable.
Last St Change	Time passed since the route entered its current state.
Next St Change	Time that the route must remain stable in order to change state from suppressed to monitored, or from monitored to ignored.

Examples To find out which routes are currently suppressed, use the command:

```
sh bgp dam rou
```

Related Commands
create bgp damping parameterset
disable bgp damping
enable bgp damping
show bgp damping

show bgp import

Syntax

```
SHow BGP IMPort
```

Description This command displays information about the BGP import entries present in the router (Figure 27-32, Table 27-18).

Figure 27-32: Example output from the **show bgp import** command

```
BGP import entries

Proto      Route map
--------------------------
OSPF       ospf_proto_map
RIP        rip_proto_map
--------------------------
```

Table 27-18: Parameters in output of the **show bgp import** command

Parameter	Meaning
Protocol	The routing protocol whose routes are to be imported into BGP; either Interface, OSPF, RIP, or Static.
Route map	The name of the route map used to filter routes and set attributes for routes imported into BGP.

Examples To show the BGP import entries on this router, use the command:

```
sh bgp imp
```

Related Commands
add bgp import
delete bgp import
set bgp import
show ip routemap in Chapter 28, Filtering IP Routes

show bgp memlimit

Syntax `SHow BGP MEMlimit`

Description This command displays the percentage of system memory that BGP and IP routing is limited to, and their combined current actual memory use (Figure 27-33).

Figure 27-33: Example output of the **show bgp memlimit** command

```
BGP Memory Limit:  85%,   Actual Use:  0%
```

Example To display the amount of memory BGP and IP routing is currently using, and its limit, use the command:

```
sh bgp mem
```

Related Commands
set bgp backoff
set bgp memlimit
show bgp memlimit scan
show buffer in Chapter 4, Configuring and Monitoring the System

show bgp memlimit scan

Syntax `SHow BGP MEMlimit SCAN`

Description This command displays information about the freelists that are registered to a given module. This output is useful to display a detailed state of BGP and IP routing memory use, at a given moment in time. (Figure 27-34, Table 27-19 on page 27-123).

Figure 27-34: Example output of the **show bgp memlimit scan** command

```
BGP Memory Limit:  85%,   Actual Use:  57%
Module Freelist Stats: moduleId = 5
module buffer use:      18686
module percent use:     34%
         list  unitSize     freeUsed  buffersUsed
-------------------------------------------------
     00f33c6c        84            0            0
     00f46b6c        12            0            0
     00f46024        88            0            0
     00f33b04        68            0            0
     00f33c9c        48       104874         4996
     00f46c78        12            0            0
     00f4f0fc        32       196984         6156
     00f557e4       236       104874        26220
-------------------------------------------------
Module Freelist Stats: moduleId = 103
module buffer use:      31011
module percent use:     57%
         list  unitSize     freeUsed  buffersUsed
-------------------------------------------------
     00d0caf4        24            0            0
     00d0ca94        12            0            0
     00d0cc40        32            0            0
     00d0cff4        32            0            0
     00d0ccc8         8          155            1
     00d0ca64       512            0            0
     00d0cbb0         8            0            0
     00d0cf14         8        19439           76
     00d0cc10         8            0            0
     00d0c7e0        64        21643          677
     00d0cbe0       524            0            0
     00d0cf9c       116          180           11
     00d0cdf8        36        19262          338
     00d0c810        20            0            0
     00d0ce28        16            0            0
     00d0cf44        52       196980         5051
     00d0cc98        40       314616         6169
     00d0bc88        20            0            0
     00d0d024        20            0            0
     00d0b860        40            2            1
     00d0bc58      1012            2            1
     00d0d07c        40            0            0
     .
     .
     .
```

Table 27-19: Parameters in output of the **show bgp memlimit scan** command

Parameter	Meaning
BGP Memory Limit	Percentage of system memory that limits BGP and IP routing.
Actual Use	Percentage of memory BGP and IP routing currently uses.
Module Freelist Stats	Statistics relating to the freelists for each module. Freelists divide memory buffers into small segments to increase efficiency of memory use.
moduleId	ID number of the software module. Module 5 displays only the IP routing details, while module 103 is a combined list of BGP and IP routing details.
module buffer use	Number of buffers currently in use by the module. Note that the buffers listed in module 5 are a subset of the buffers listed in module 103.
module percent use	Number of buffers currently used by the module, as a percentage of the total number of system buffers.
list	Freelists (as a hexadecimal address) registered to the module.
unitSize	Number of bytes each freelist segment uses.
freeUsed	Number of segments of the freelist currently used by the module.
buffersUsed	Number of memory buffers the freelist is currently using.

Example To display the detailed state of current BGP memory use, use the command:

```
sh bgp mem scan
```

Related Commands

set bgp backoff
set bgp memlimit
show bgp memlimit
show buffer in Chapter 4, Configuring and Monitoring the System

show bgp network

Syntax `SHow BGP NETwork`

Description This command displays information about the BGP network entries configured in this router (Figure 27-35, Table 27-20).

Figure 27-35: Example output from the show bgp network command

```
BGP network entries

Prefix                Route map
-------------------------------------
192.168.248.0/21      network_map
192.168.16.0/21       -
-------------------------------------
```

Table 27-20: Parameters in output of the **show bgp network** command

Parameter	Meaning
Prefix	Prefix for this network entry. This is the prefix that BGP advertises for this network as long as a route that matches this prefix exactly is present in the BGP routing table.
Route map	Name of the route map that is used to filter routes and set the BGP attributes for this network entry.

Examples To display information about BGP networks, use the command:

```
sh bgp net
```

Related Commands

add bgp network
delete bgp network
show bgp
show ip routemap in Chapter 28, Filtering IP Routes

show bgp peer

Syntax `SHow BGP PEer[=ipadd]`

Description This command displays:

- summary information about all BGP peers if you do not specify a peer address (Figure 27-36, Table 27-21)
- detailed information about a peer, if you specify the IP address of the peer (Figure 27-37 on page 27-126, Table 27-22 on page 27-127). Addresses are specified in dotted decimal notation.

Figure 27-36: Example summary output from the **show bgp peer** command

```
BGP peer entries
Peer          State     AS     InMsg  OutMsg  Template  Role
------------------------------------------------------------------
192.168.2.254 Estab     12345  23456  3245    -         non-client
192.168.3.16  Idle(D)   123    2      3       2         client
```

Table 27-21: Parameters in output of the **show bgp peer** command

Parameter	Meaning
Peer	IP address of the BGP peer.
State	BGP peer state, one of: • Idle • Idle (D)—Idle and also disabled • Connect • Active • OpenSent • OpenConf—OpenConfirm • Estab—Established For more information about the states, see "BGP Operation" on page 27-5.
AS	Number of the autonomous system to which this peer belongs.
InMsg	Number of messages received from this peer since the TCP connection opened.
OutMsg	Number of messages sent to this peer since the TCP connection opened.
Template	ID number of the peer policy template that provides the peer with its settings.
Role	Whether the peer is an EBGP peer of the router, or for IBGP peers, whether the peer is a client or non-client peer for route reflection. For information about route reflection, and client and non-client peers, see "How to Improve IBGP Scalability" on page 27-33.

Figure 27-37: Example output from the **show bgp peer** command for a specific peer

```
Peer ................ 192.168.10.1
Description ......... -
State ............... Idle
Policy Template ..... 4
  Description ....... Test Template 1
Private AS filter ... Yes
Remote AS ........... 3
BGP Identifier ...... 172.20.25.2
Routes learned ...... 15
Authentication ...... None
  Password .......... -
Fast Fall-Over ...... ENABLED
Default originate ... DISABLED
Role ................ Client
Connect retry ....... 120s
Hold time ........... 90s
Keep alive .......... 30s
Min AS originated ... 15
Min route advert .... 30
Local Interface ..... Not defined
Capability Matching.. Strict

Filtering
  In filter ......... -
  In path filter .... -
  In route map ...... -
  Out filter ........ -
  Out path filter ... -
  Out route map ..... -

Max prefix .......... OFF
External hops ....... 1 (EBGP multihop disabled)
Next hop self ....... No
Send community ...... No
Messages In/Out ..... 0/0
Debugging ........... -
  Device ............ -

Capabilities ......... Route Refresh

Established transitions ........... 0

Session Message counters:
  inOpen ........................... 0       outOpen ........................... 0
  inKeepAlive ...................... 0       outKeepAlive ...................... 0
  inUpdate ......................... 0       outUpdate ......................... 0
  inNotification ................... 0       outNotification ................... 0
  inRouteRefresh ................... 0       outRouteRefresh ................... 0

Total Message counters:
  inOpen ........................... 0       outOpen ........................... 0
  inKeepAlive ...................... 0       outKeepAlive ...................... 0
  inUpdate ......................... 0       outUpdate ......................... 0
  inNotification ................... 0       outNotification ................... 0
  inRouteRefresh ................... 0       outRouteRefresh ................... 0
```

Table 27-22: Parameters in output of the **show bgp peer** command for a specific peer

Parameter	Meaning
Peer	IP address of the BGP peer.
Description	Description of the peer if it has one.
State	BGP peer state, one of: • Idle • Idle (D)—Idle and also disabled • Connect • Active • OpenSent • OpenConfirm • Established For more information about the states, see "BGP Operation" on page 27-5.
Policy Template	ID number of the peer policy template that provides the peer with its settings.
Description	Description of the peer policy template if it has one.
Private AS filter	Whether private AS numbers (from 64512 to 65535) are stripped from the AS PATH attribute on update messages the router sends to the peer. "Yes" indicates private AS numbers are stripped. "No" indicates they are not.
Remote AS	Number of the autonomous system to which this peer belongs.
BGP identifier	The ID this router uses to identify itself to the peer. For more information, see "How to Set the IP Address that Identifies the Router" on page 27-40.
Routes learned	The number of routes that the router has learned from this peer.
Authentication	Authentication type used for communication with this BGP peer, or None if the connection is not using authentication.
Password	Password for this peer if the connection uses authentication.
Fast-Fallover	Whether fast fallover is enabled on the peer. Fast fallover improves convergence when topology changes, by resetting the BGP session as soon as the router's interface to the peer goes down.
Default originate	Whether BGP advertises the default route (0.0.0.0/0) to this peer, when the router's BGP routing table contains the default route.
Role	Whether the peer is an EBGP peer of the router, or for IBGP peers, whether the peer is a client or non-client peer for route reflection. For information about route reflection, and client and non-client peers, see "How to Improve IBGP Scalability" on page 27-33.
Connect Retry	The time interval for retrying the initial TCP connection to this peer in the event of a connection failure.
Hold time	The configured and actual hold times for this peer. The actual hold time is the lower of the configured hold times of the peer and this router.
Keep alive	The configured and actual keepalive times for this peer. The actual keepalive time is set by the actual hold time in such a way that the ratio of actual keepalive to hold time is the same as the ratio of configured keepalive to hold time.

Table 27-22: Parameters in output of the **show bgp peer** command for a specific peer (cont)

Parameter	Meaning
Min AS originated	Minimum time between advertisements of routes that originate in this autonomous system.
Min route advert	Minimum time between advertisements of routes that originate outside this autonomous system.
Filtering	Settings for inward and outward filtering of routing information via BGP.
In filter	Traffic filter used for filtering incoming routes from this peer.
In path filter	AS path filter used for filtering incoming routes from this peer.
In route map	Route map used for filtering incoming routes from this peer.
Out filter	Traffic filter used for filtering outgoing routes to this peer.
Out path filter	AS path filter used for filtering outgoing routes to this peer.
Out route map	Route map used for filtering outgoing routes to this peer.
Max prefix	Maximum number of route prefixes that may be received from this peer, and the action taken when this number is exceeded. The action is WARNING or TERMINATE.
External hops	Number of hops that can be used to reach this peer when it is an EBGP peer. Having this number exceed 1 allows multihop EBGP.
Next hop self	Whether this router advertises to this peer that the next hop for all routes is itself.
Send community	Whether this router sends the community attribute in the path attributes of update messages.
Messages In/Out	Number of incoming/outgoing BGP messages from/to this peer.
Debugging	Debugging types enabled for this peer: msg, state, update, and all.
Device	Device where debugging output is sent.
Local Interface	The local interface. In certain circumstances, the router uses this address as the source for BGP packets it generates and sends to the peer. For a description of when the router uses the local interface, see "How to Set the IP Address that Identifies the Router" on page 27-40.
Capability Matching	How the local BGP speaker (the router) compares this remote peer's capabilities with its own capabilities. **Strict** indicates that the local speaker only establishes a session to the peer if the local speaker supports all the capabilities that the remote peer advertises in its Open message. **Loose** indicates that the local speaker and the peer establish a session as long as the remote peer supports at least the IPv4 address family and IPv4 unicast capabilities.
Capabilities	Extra capabilities negotiated between the peer and the router. "Route Refresh" indicates that the router automatically sends route refresh messages to the peer and process route refresh messages from the peer. Route refresh messages request a new update message, and are used after a BGP peer has been modified, to reset its routes.
Established transitions	Number of times the peer session has become established (entered the Established state).

Table 27-22: Parameters in output of the **show bgp peer** command for a specific peer (cont)

Parameter	Meaning
Message Counters	**Session Message Counters** give the number of messages received or sent for the most recently established session with the peer. If the peer session is torn down at any time, the session counters are reset and accumulate from zero when a new session is established. **Total Message Counters** give the number of messages received or sent for all sessions ever established with the peer. For more information about messages, see "BGP Operation" on page 27-5.
InOpen	Number of *open* messages received from this peer. BGP peers use open messages to identify themselves to each other and negotiate settings.
OutOpen	Number of *open* messages sent to this peer. BGP peers use open messages to identify themselves to each other and negotiate settings.
InKeepAlive	Number of keepalive messages received from this peer. Keepalive messages maintain the BGP session when the peer has not needed to send update messages.
OutKeepAlive	Number of keepalive messages sent to this peer. Keepalive messages maintain the BGP session when the router has not needed to send update messages.
InUpdate	Number of update messages received from this peer. BGP peers use update messages to inform each other of route changes.
OutUpdate	Number of update messages sent to this peer. BGP peers use update messages to inform each other of route changes.
InNotification	Number of notification messages received from this peer. BGP peers use notification messages to inform each other of errors.
OutNotification	Number of notification messages sent to this peer. BGP peers use notification messages to inform each other of errors.
InRouteRefresh	Number of route refresh messages received from this peer.
OutRouteRefresh	Number of route refresh messages sent to this peer.

Example To display summary information for all BGP peers, use the command:

```
sh bgp pe
```

To display detailed information for the BGP peer 192.168.1.1, use the command:

```
sh bgp pe=192.168.1.1
```

Related Commands

add bgp peer
delete bgp peer
set bgp peer
show bgp
show ip routemap in Chapter 28, Filtering IP Routes

show bgp peertemplate

Syntax `SHow BGP PEERTemplate[=1..30]`

Description This command displays information about all BGP peer policy templates, or about the specified BGP peer policy template (Figure 27-38, Table 27-23).

The **peertemplate** parameter specifies the identification number of the BGP peer policy template. If you do not specify a value, information is displayed for all BGP peers.

Figure 27-38: Example output from the **show bgp peertemplate** command

```
BGP Peer Template Information
----------------------------------------------------------------
Template.............. 1
Description ......... -
Role ................. Client
Connect retry ....... 120s
Hold time ........... 90s
Keep alive .......... 30s
Min AS originated ... 15
Min route advert .... 30

Filtering
  In filter ......... -
  In path filter .... -
  In route map ...... -
  Out filter ........ -
  Out path filter ... -
  Out route map ..... -

Max prefix .......... OFF
Next hop self ....... No
Send community ...... No

Private AS Numbers .. Don't Filter
----------------------------------------------------------------
```

Table 27-23: Parameters in output of the **show bgp peertemplate** command

Parameter	Meaning
Template	ID number of the template.
Description	Description for peers that use the template.
Role	Whether peers that use the template are EBGP peers of the router, or for IBGP peers, whether the peers are client or non-client peers for route reflection. For information about route reflection, and client and non-client peers, see "How to Improve IBGP Scalability" on page 27-33.
Connect Retry	Interval for retrying the initial TCP connection to peers that use the template in the event of a connection failure.
Hold time	The configured hold time for peers that use the template. The actual hold time is the lower of the configured hold times of the peer and this router.

Table 27-23: Parameters in output of the **show bgp peertemplate** command (cont)

Parameter	Meaning
Keep alive	The configured keepalive time for peers that use the template. The actual keepalive time is set by the actual hold time in such a way that the ratio of actual keepalive to hold time is the same as the ratio of configured keepalive to hold time.
Min AS originated	Minimum seconds between advertisements from the router to peers that use the template for routes that originate in this autonomous system.
Min route advert	Minimum seconds between advertisements from the router to peers that use the template for routes that originate outside this autonomous system.
Filtering	Settings for inward and outward filtering of routing information via BGP.
In filter	Traffic filter used for filtering incoming routes from peers that use the template.
In path filter	AS path filter used for filtering incoming update messages from peers that use the template.
In route map	Route map used for filtering incoming routes or update messages from peers that use the template, and/or for setting their attributes.
Out filter	Traffic filter used for filtering outgoing routes to peers that use the template.
Out path filter	AS path filter used for filtering outgoing update messages to peers that use the template.
Out route map	Route map used for filtering outgoing routes or update messages to peers that use the template, and/or for setting their attributes.
Max prefix	Maximum number of route prefixes that may be received from peers that use the template, and the action taken when this number is exceeded. The action is WARNING or TERMINATE.
Next hop self	Whether this router advertises to peers using the template that the next hop for all routes is itself.
Send community	Whether this router sends the community attribute in the path attributes of update messages to peers that use the template.
Private AS Numbers	Whether private AS numbers (from 64512 to 65535) are stripped from the AS PATH attribute on update messages the router sends to peers that use the template. "Filter" indicates private AS numbers are stripped. "Don't Filter" indicates they are not.

Example To display the settings of peer template 1, use the command:

```
sh bgp peert=1
```

Related Commands

add bgp peer
add bgp peertemplate
delete bgp peertemplate
set bgp peer
show bgp

show bgp route

Syntax

```
SHow BGP ROUte[=prefix]
   [COMmunity={INTernet|NOAdvertise|NOExport|
   NOEXPORTSubconfed|aa:xx}[,...]}] [PEer=ipadd]
   [REGexp=aspathregexp]
```

Description

This command displays information about some or all routes in the BGP routing table (Figure 27-39 on page 27-133 and Table 27-24 on page 27-133).

Parameter	Description	
ROUte	The network prefix of the routes to display. The *prefix* is an IP address in dotted decimal notation, optionally followed by the CIDR mask. The router displays all routes that match the prefix or that are subnets of the prefix. If you do not specify a prefix, the router displays all BGP routes that match the **regexp** and **community** specified. Default: no default	
COMmunity	A community name or number. The router only displays routes with this value of the community attribute. Note that if you specify a community, routes that do not contain a community attribute are not displayed. Default: no default	
	INTernet	The community of routes that can be advertised to all BGP peers.
	NOExport	The community of routes that must not be advertised outside a BGP confederation boundary (a standalone autonomous system that is not part of a confederation should be considered a confederation itself).
	NOAdvertise	The community of routes that must not be advertised to other BGP peers.
	NOEXPORTSubconfed	The community of routes that must not be advertised to external BGP peers (this includes peers in other members' autonomous systems inside a BGP confederation).
	aa:*xx*	The number of a community. ***aa*** and ***xx*** are both integers in the range 0 to 65534. ***aa*** is the AS number. ***xx*** is a value chosen by the ASN administrator.
PEer	The IP address of the peer. If you specify a peer, the router only displays routes that it learned from that peer. If you specify the router's router ID, it displays all locally originated routes. Default: no default	

Parameter (cont)	**Description** (cont)
REGexp	An AS path regular expression. The router only displays routes with an AS path attribute that matches the regular expression. Regular expressions are a list of one or more AS numbers separated by spaces. To match from the first number in the list, start the expression with the ^ character. To match the last number, end with the $ character. If the expression contains spaces, surround it with double quotes. For more information about valid syntax, see "Creating AS Path Lists for BGP Routes" on page 28-9 of Chapter 28, Filtering IP Routes. For example: • **regexp="23334 45634 88988"** displays any route with a path containing these numbers • **regexp="^23334 45634 88988$"** displays any route with that exact path • **regexp=^23334** displays any route with a path beginning with 23334 Default: no default

Figure 27-39: Example output from the **show bgp route** command

```
BGP route table
Flags: >=Best route for the given prefix, *=Unreachable next hop, W=Withdrawn
       a=Aggregate route, s=Aggregate Suppressed, D=Damped
       Learned from: L=Local, e=eBGP Peer, i=iBGP Peer, c=Confederate Peer
--------------------------------------------------------------------------------
Fl  Prefix              Next hop         Origin      MED         Local pref
    Path
    Originator          Cluster List
--------------------------------------------------------------------------------
  > 192.123.1.0/24      192.168.2.1      IGP         -           100
    SEQ 100;SEQ 567 4345 4234 37623 23445 452 456 2663 8664 6221 24256
    7453 134;SET 256 784 2134 3456;
  e -                   -
  > 172.168.28.0/22     192.168.9.1      IGP         -           100
    SEQ 100;
  i -                   -
    172.168.28.0/22     192.168.2.1      IGP         14          100
    SEQ 100;
  e -                   -
*   172.168.97.0/24     172.192.3.4      IGP         -           130
    EMPTY
  i 172.192.3.18        172.192.3.15     172.192.3.54
  > 192.168.9.0/24      192.168.17.1     INCOMPLETE  -           100
    EMPTY
  L -                   -
--------------------------------------------------------------------------------
```

Table 27-24: Parameters in output of the **show bgp route** command

Parameter	**Meaning**	
Fl	identifier (flag) that indicates the route's current state, or how BGP discovered the route.	
	>	BGP has selected this route as the best route to this destination, and has added this route to the IP routing table.
	*	BGP is excluding this route from selection, as the route's next hop is unreachable.

Table 27-24: Parameters in output of the **show bgp route** command (cont)

Parameter	Meaning	
	W	A peer has withdrawn this route. When BGP route flap damping is enabled, BGP continues to show these withdrawn routes in the table.
	a	This route is an aggregate of other routes in the table.
	s	BGP is excluding this route from being advertised to other peers, because an aggregate route has auto-summary enabled. BGP can still select this as the best route to the destination for this router, however it advertises the parent aggregate entry to other peers.
	D	BGP is suppressing this route because route flapping is occuring. BGP will continue suppressing the route until the route becomes more stable.
	L	BGP learned this route through the **add bgp import**, **add bgp network** or **add bgp aggregation** command.
	e	BGP learned this route from an EBGP neighbour.
	i	BGP learned this route from an IBGP neighbour.
	c	BGP learned this route from a confederate neighbour.
	?	BGP cannot determine how it learned this route. This indicates that an internal error has occured.
Prefix	Network prefix for this route.	
Next hop	IP address of the next hop for this route. A next hop of 0.0.0.0 indicates it is an interface route.	
Origin	Origin attribute for this route; one of IGP, EGP, or Incomplete.	
MED	Multi Exit Discriminator attribute for this route. "0" indicates that BGP received a MED of 0, whereas "-" indicates that BGP received no MED. The router's implementation of BGP treats "0" and "-" identically.	
Local Pref	Local preference attribute for this route. The default value is 100.	
Path	AS path attribute for this route. This consists of the AS sequences, sets, confederation sequences and confederation sets.	
Originator	Router ID of the first IBGP peer from which BGP learned this route, if any. Displays only when BGP is using route reflection.	
Cluster List	List of the router reflection clusters that have forwarded this route, if any. Displays only when BGP is using route reflection.	

Examples

To display the BGP route table for routes that pass through AS 1234, use the command:

```
sh bgp rou reg=1234
```

Related Commands

show bgp
show bgp aggregate
show bgp import
show bgp network
show bgp peer