Chapter

8

Border Gateway Protocol (BGP)

JNCIA EXAM OBJECTIVES COVERED IN THIS CHAPTER:

- ✓ Define the basic operation and functionality of BGP
- ✓ Describe the different BGP neighbor states
- ✓ Define the functions of BGP packet types
- ✓ Define the functions of BGP attributes
- ✓ Identify the steps of the BGP Route Selection Algorithm
- ✓ Describe the default action for BGP route advertisements to EBGP and IBGP peers
- ✓ Describe the steps required to configure BGP
- ✓ Identify CLI commands used to monitor and troubleshoot a BGP network

This chapter explores the Border Gateway Protocol (BGP). This routing protocol is used extensively in the Internet to connect ISP networks. We start with an examination of how BGP forms peer relationships with neighboring routers. This is followed by a discussion of the packets used to exchange information and how a BGP router uses the route information. After exploring some BGP route attributes, we finish by looking at how BGP is configured within the JUNOS software.

Overview of BGP

BGP was created to achieve some specific goals. First, it needed to support the meshlike connectivity of ISP networks. It also required extensive policy controls to enforce the administrative policies of each ISP. It needed the ability to reliably transmit routing information between BGP peers. Finally, the protocol required the ability to scale route advertisements beyond a few thousand routes. Version 4 of the protocol accomplishes each of these goals.

Network Connectivity

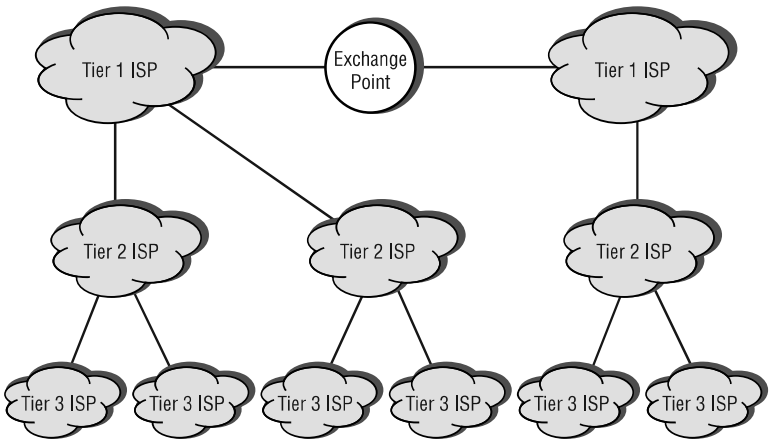
In the early days of the Internet, connectivity between networks was regimented and hierarchical. For example, the ARPANET dictated a single backbone that provided transport services for each connected network. The Exterior Gateway Protocol (EGP) was used for inter–Autonomous System (AS) communications. It provided no loop prevention and sent its entire routing table in regular broadcasted updates. At the time, EGP satisfied the requirements of the Internet. As the ARPANET was disbanded and the Internet grew, this hierarchy grew into the concept of a tiered network design. Small networks (Tier 3 ISPs) connected upstream to larger regional networks (Tier 2 ISPs). The regional systems would, in turn, connect to the major ISPs (Tier 1 providers). Regional exchanges points were established to allow the Tier 1 ISPs to interconnect among themselves. This basic concept is shown in Figure 8.1.

The purpose of the Internet also changed during this timeframe from a research-type network to a commercial data-transport system. This transition forced many networks into providing better and faster service to their customers, necessitating a change in the design. Some ISPs began to connect to multiple upstream providers. Most Tier 1 ISPs connected to other peers outside the exchange points and in multiple locations.

The Internet growth also meant that more reachable prefixes were advertised to the network on a regular basis. This increased routing traffic as well as a more meshlike structure spelled the end of EGP. It simply was not designed for this type of environment. BGP, on the other hand,

was created for just this situation. It provides loop prevention through an attribute called the AS Path, which is a collection of AS numbers through which a particular route has passed.

FIGURE 8.1 The tiered Internet design



An Autonomous System, in reality, is a set of routers under the control of a single administrative domain. An individual domain may have more than one AS number assigned to it. On the other hand, a single AS number can belong only to a single domain. For our purposes, we assume that an AS means a single number assigned to a single administrative domain.

The use of this attribute also leads to a common description of BGP as a *path-vector protocol*. In a distance-vector network, such as the Routing Information Protocol (RIP), each router knows how far away a route is (the distance) and the direction to send a packet for that route (the vector). In a BGP network, on the other hand, each router knows the networks a route has traversed (the path) and the direction to send a packet for the route (the vector).

Policy Control

The use of the Internet for commercial traffic also requires that ISPs have extensive control over route advertisements and traffic patterns. The requirements for this control can come in many forms. For example, a government agency might dictate that all traffic originating and terminating within a specific country must use networks only within that country. If this set of networks is not naturally the best path toward the destination, the ISPs need a method for changing route attributes to make it the best path.

Another example might occur when two ISPs connect to each other on a private link outside an exchange point. In this situation, one of the networks might desire only certain routes from its peer. These routes could be the specific customers of the Tier 1 ISP, or a subset of the customer routes.

Another use of policy control occurs when some network links are not used for primary data traffic flows. Suppose a Tier 3 ISP purchases connectivity to a second upstream peer for use as a backup link. This link should be used only when its primary upstream link is not operational. The Tier 3 ISP could advertise its routes on this backup link but make them appear less attractive to the Internet at large by altering certain attributes. This dual advertisement provides for a faster fail-over of traffic should the primary link fail.

BGP accounts for each of these situations. It allows for explicit inbound and outbound policy controls on a per-peer basis. These controls permit an ISP to block or advertise particular routes. Each ISP may also alter a number of attributes to change the attractiveness of a particular route to its peers.

The basic BGP attributes are discussed in the section "BGP Attributes," later in this chapter.

Reliable Transport

The size of the modern-day Internet routing table (over 100,000 routes) and the need for a high level of availability require the Internet routing protocol to guarantee its transmissions. This guarantee is not for actual user data traffic but for the protocol transmissions itself. BGP accomplishes this goal by utilizing the Transmission Control Protocol (TCP) as its underlying transport mechanism. This greatly differs from some Interior Gateway Protocols (IGPs), such as Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS), where the reliability is built into the protocol itself. When two BGP peers establish a session on *TCP port 179*, they get a connection-oriented and reliable stream of data between them. The peers can then rely on TCP for the following services:

Acknowledgments After sending a segment, a retry timer starts decrementing until a receipt acknowledgment is received from the other end of the connection. When the timer reaches 0, the segment is retransmitted. The far-end acknowledgment is actually delayed up to 1 second to determine if any data should be sent along with the acknowledgment.

Segmentation The BGP data is segmented, if necessary, into smaller sizes for transmission across the network.

Checksums A checksum is maintained on both the TCP header and BGP data to ensure an error-free transmission. Should the received checksum differ from the advertised value, the segment is discarded. No acknowledgment is sent to the source, and the segment is retransmitted.

Data sequencing TCP sequence numbers allow the receiving peer to reorder the BGP data in the event of an out-of-sequence receipt.

Flow control Each BGP peer advertises its available buffer space to allow the far end of the session to send only a specific amount of data.

Real World Scenario

Is BGP Really a Routing Protocol?

This question can often start a very heated discussion between two network engineers—in large part because there is no clear answer to the question. Your particular answer is defined by your beliefs and experiences.

On one hand are the people who answer yes. After all, the end result of using BGP is the advertisement of IP routes. A router places these routes into the routing table after determining which version of the route is the best. The router then forwards user data packets based on the table's information.

People on the other side of the issue consider BGP to be an application of IP. Traditional routing protocols have their own protocol number. As such, they are part of the IP environment. Since BGP uses TCP for its transmissions, it actually performs its job as an application, like Telnet. The difference between the applications is the data transmitted between the end hosts. Telnet provides character-based terminal access to the far-end host. BGP, on the other hand, provides route knowledge to the far-end host. There is simply no difference.

We won't try to sway your vote to one side or the other. We'll simply attempt to discuss the facts of what BGP does and how it does it. The rest is up to you.

Routing Table Scalability

The rapid growth of the Internet routing table requires any inter-AS protocol to scale effectively. The original EGP specification required that each router advertise its entire routing table space on a regular cycle. This is similar to RIP in its operation and carries with it the scalability issues of a traditional distance-vector protocol. BGP uses a route advertisement mechanism that is more comparable to OSPF or IS-IS.

When two BGP peers establish a connection, they initially advertise their entire routing knowledge to each other. After this initial exchange, updates occur only on an as-needed basis. Updates are sent only when new information is learned or existing information is no longer valid. This change-based system provides the best possible scalability for the Internet's routing protocol.

When Will BGP Be Updated?

The current version of BGP is version 4, originally defined in 1995. No newer versions of the protocol have been introduced since then, due in large part to the adaptability of the protocol. As network needs have changed, the base specification has been modified to meet the new requirements. This means that no new version of BGP is needed for the foreseeable future.

A number of Requests for Comment (RFC) define the current BGP specification. These RFCs include:

- RFC 1771, "A Border Gateway Protocol (BGP-4)"
- RFC 1772, "Application of the Border Gateway Protocol in the Internet"
- RFC 1966, "BGP Route Reflection: An Alternative to Full-Mesh I-BGP"
- RFC 1997, "BGP Communities Attribute"
- RFC 2270, "Using a Dedicated AS for Sites Homed to a Single Provider"
- RFC 2385, "Protection of BGP Sessions through the TCP MD5 Signature Option"
- RFC 2439, "BGP Route Flap Damping"
- RFC 2842, "Capabilities Advertisement with BGP-4"
- RFC 2858, "Multiprotocol Extensions for BGP-4"
- RFC 2918, "Route Refresh Capability for BGP-4"
- RFC 3065, "AS Confederations for BGP"

Additionally, the JUNOS software supports various Internet Engineering Task Force (IETF) drafts for BGP. Two of these include:

- "BGP Extended Communities Attribute", IETF draft-ramachandra-bgp-ext-communities-04.txt
- "Graceful Restart Mechanism", IETF draft-ietf-idr-restart-01.txt

To locate IETF drafts, go to `http://www.ietf.org/ID.html` and search for a draft or examine a complete index of all current drafts.

Theory of Operation

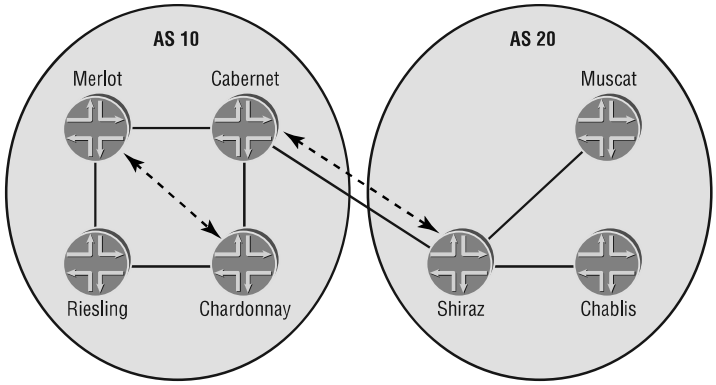
BGP advertises routing knowledge between two routers. The type of relationship between those routers is determined by whether they are in the same AS. The establishment of this relationship and the actual advertisement of routes are controlled by various BGP packet types. The local router makes a routing decision based on the received routing knowledge. These decisions are then advertised to other BGP routers in the network. The routing decisions are based on a number of variables, including various route attributes and local policy controls.

We cover each of these major aspects in more detail throughout this section. Let's begin with a look at how BGP forms relationships with its neighboring routers.

Peers

BGP exchanges its routing information between two routers, called *peers* or *neighbors*. This connection is logical in nature and relies on the establishment of a TCP session between the peers. The session is established across a direct physical link or a number of intermediate links. Figure 8.2 shows each of these situations.

FIGURE 8.2 BGP peer relationships



The Cabernet and Shiraz routers are joined by a direct physical connection. This connectivity implies that IP reachability is easily achieved between the two peers and that TCP can establish its session. On the other hand, the Merlot and Chardonnay routers are connected only via an intermediate router. In this situation, IP reachability between the peers must be supplied by some other means. Often this is handled by the IGP within the AS, but a static route is also a feasible solution.

Figure 8.2 also shows the different forms of logical connections used by BGP. One option connects two routers in different AS networks, like the Cabernet-Shiraz session. The second option is for two routers in the same AS to establish a session; this is represented by the Merlot-Chardonnay connection. While TCP sessions are established based on the IP reachability between two peers, BGP uses each type of session in a different manner.

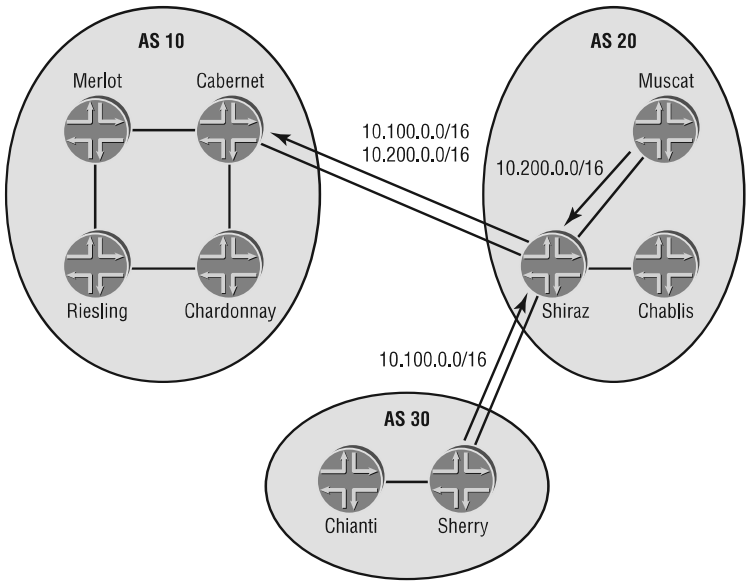
External BGP Sessions

When two BGP routers are in different AS networks, the session between them is considered an *external BGP (EBGP)* connection. By default, an EBGP connection is formed between directly connected peers. This requirement is enforced by setting the time-to-live (TTL) of the IP packet to 1, thereby not permitting an intermediate router to forward the BGP packet.

Once the EBGP session is established, the two peers can begin to exchange routing knowledge with each other. All active BGP routes learned from other EBGP sessions are advertised. In addition, all active BGP routes learned from internal BGP peers are advertised.

Figure 8.3 shows the default EBGP route advertisements. The 10.100.0.0 /16 route is advertised from the Sherry router to Shiraz via an EBGP session. The 10.200.0.0 /16 route is advertised to Shiraz by the Muscat router, which is an internal peer. Both routes are currently active in Shiraz's routing table and are advertised to Cabernet using an EBGP advertisement.

FIGURE 8.3 EBGP route advertisements



Internal BGP Sessions

The connection of two BGP routers within the same AS is called an *internal BGP (IBGP)* connection. Unlike the EBGP variety, there is no requirement for physical connectivity between IBGP peers. The TTL of the BGP packets is set to 64 to allow for connectivity across an AS. In fact, a great majority of IBGP sessions are between routers that are not directly connected.

Real World Scenario

Internal BGP Design Considerations

Most ISP networks run BGP on all routers in their AS. Of course, any customer-facing router runs the protocol to announce routing information, but this design rule holds true for core routers as well. One reason for this setup is the avoidance of black-holing user data traffic. Let's look at a simple example.

Suppose that AS 65000 consists of three routers (A, B, and C) connected together in a chain; A connects to B, which connects to C. AS 65000 uses OSPF as its IGP for internal reachability. Router A has an external BGP session with AS 64777, and Router C has an EBGP session with AS 64888. Additionally, Routers A and C have an IBGP session between them. The administrators of AS 65000 choose to not have Router B participate in BGP because of a concern over router resources. It receives a default 0.0.0.0 /0 route from both Router A and Router C for reachability to the Internet at large.

So far, so good. Each router in the AS has usable routes in its routing table for Internet destinations. The problem occurs when a user connected to Router B wants to connect to `www.juniper.net`. The data packet arrives at Router B, and a routing table lookup is performed. Router B finds two copies of the default route and must choose one. Let's assume it selects Router C and forwards the packet. Router C then performs a route lookup and finds an explicit route to `www.juniper.net` through its IBGP peer—Router A. Router C forwards the packet to Router A, using Router B as the physical path to its peer.

You may already see the problem developing, but let's finish the process. Router B receives a data packet destined for `www.juniper.net`. It doesn't realize that it just forwarded this packet a second ago and performs a route lookup. It once again finds two default routes and forwards the packet back to Router C. Congratulations; we have just established a routing loop!

One obvious solution to this problem is establishing an IBGP session from Router B to both Router A and Router C. Once established, Router B has an explicit route for `www.juniper.net`. It no longer forwards the user data packets to Router C, but to Router A instead. The routing loop has been averted.

IBGP peers rely on the IGP knowledge within the AS network. In general, the TCP session between IBGP peers across the network uses the IP routing tables of intermediate nodes to establish itself. More specifically, this session is established using the loopback addresses of the peers for stability and resiliency. This common practice allows the IBGP session to remain operational in the event of a network outage.

Once the IBGP session is established, routes are exchanged between the peers. By default, only active BGP routes learned from EBGP peers are advertised across an IBGP session.

The default IBGP advertisement rules are shown in Figure 8.4. The Cabernet and Riesling routers are IBGP peers. Cabernet is learning both the 10.100.0.0 /16 and 10.200.0.0 /16 routes from Shiraz, an EBGP peer. Both of these routes are then readvertised to Riesling across the IBGP session. The Riesling router is also learning about the 172.30.1.0 /24 route from an IBGP peer—Chardonnay. This route is not advertised to the other AS 10 routers because it violates the advertisement rules for IBGP peers.

The restriction on advertising an IBGP-learned route to another IBGP peer causes an interesting problem within AS 20, as seen in Figure 8.5.

FIGURE 8.4 IBGP route advertisements

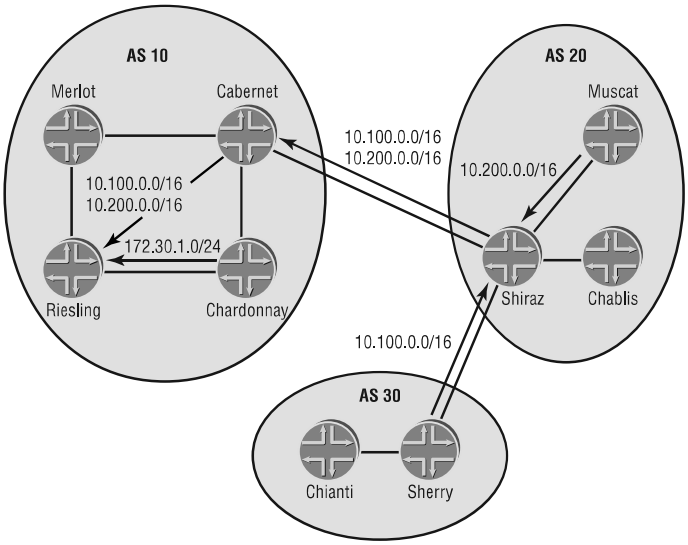
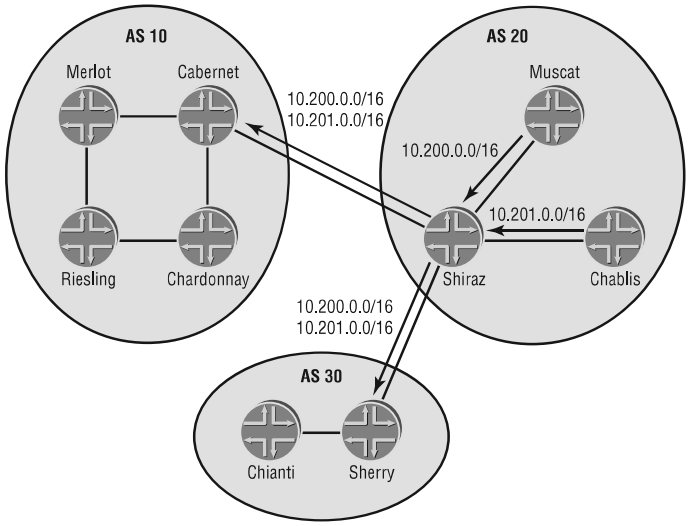


FIGURE 8.5 The IBGP full-mesh requirement



Both the Muscat and Chablis routers have an IBGP session with Shiraz, but not with each other. The 10.200.0.0 /16 route is advertised by Muscat, and the 10.201.0.0 /16 route is sent by Chablis. Shiraz receives both routes and advertises them to its EBGP peers of Cabernet and Sherry. By default, it does not advertise those routes back to its IBGP peers. This results in both Chablis and Muscat having no knowledge of other routes within the AS.

The resolution to this dilemma is the establishment of an IBGP session between Muscat and Chablis. Remember that IBGP peers do not need to be physically connected. Once established, both Muscat and Chablis advertise their routes to each other directly, in addition to advertising to Shiraz. The end result is an *IBGP full-mesh* within AS 20. This full-mesh concept is considered a best practice within an ISP network to maintain route reachability.

Real World Scenario

Why Do We Need an IBGP Full-Mesh?

In Figure 8.5, we saw a graphic need for an IBGP full-mesh due to the default advertisement rules for IBGP peers. But the question about why the advertisement rules are in place is still a valid one.

The first, and easiest, answer is because the RFC says so. If you are happy with that response, please read no further. Personally, we never find those types of answers satisfying, so let's see if we can dig a little further.

The underlying reason for the full-mesh requirement stems from the use of the AS Path attribute. In the "Network Connectivity" section earlier in this chapter, we discussed the AS Path attribute as the method BGP uses for loop prevention. Each BGP router examines this attribute when it receives a route from a peer. If the local router's AS number is listed in the attribute, then the route has already been through this AS. The local router then drops the route advertisement to prevent the formation of a routing loop.

Having said all that, we still haven't answered the question! Okay, here it is. The AS Path attribute is modified only when a route is advertised between EBGP peers, not IBGP peers. So, if BGP were to allow an IBGP router to advertise an IBGP-learned route to another IBGP peer, we could easily form a routing loop.

As an example, suppose that Cabernet, Merlot, Riesling, and Chardonnay in Figure 8.5 are all IBGP peers across the physical connections only (Cabernet and Riesling are not peers, for example). In addition, suppose we allow the IBGP peers to advertise routes to other IBGP peers. When Cabernet receives the 10.200.0.0 /16 route from Shiraz, it advertises that route to Merlot and Chardonnay. Those two routers select Cabernet's version of the route as the best and forward it to Riesling. At this point, two versions of the same route have arrived on Riesling, which must choose between them. Riesling opts for the Merlot version and forwards the route to Chardonnay, which prefers the same version. Chardonnay then forwards the route to Cabernet, which announced the route in the first place!

Cabernet now has a decision to make between the versions from Shiraz and Chardonnay. Using the version from Chardonnay installs a routing loop in AS 10. User data packets from Merlot arrive at Cabernet, which forwards them to Chardonnay. Route lookups on Chardonnay forward the packets to Riesling, which forwards them back to Merlot. Unfortunately, Cabernet doesn't have the ability to foresee this danger through its loop avoidance mechanism, the AS Path. Neither the route from Shiraz nor the route from Chardonnay lists AS 10 in the AS Path attribute. Allowing the AS 10 routers to announce routes among themselves destroys the guarantee of a loop-free topology.

So, the BGP designers decided to not allow you to shoot yourself in the foot by preventing IBGP learned routes from being advertised to other IBGP peers.

Establishing Relationships

Now that we've established the basic concepts of BGP peering, let's take a closer look at how the sessions are actually formed. BGP uses a Finite State Machine model when forming a peer relationship. Here are the possible BGP states:

`Idle` *Idle* is the initial neighbor state, in which it rejects all incoming session requests. After the BGP process starts, a TCP session is initiated to the remote peer. The local router transitions to the `Connect` state and begins to listen for a connection initiated by the remote router.

`Connect` In the *Connect* state, the local router is waiting for the TCP session to be completed. If it is successful, the local router sends an Open message to the peer and transitions to the `OpenSent` state.

Should the TCP connection attempt fail, the local router resets the ConnectRetry timer and transitions to the `Active` state.

If the ConnectRetry timer reaches 0 while the local router is in the `Connect` state, the timer is reset and another connection attempt is made. The local router remains in the `Connect` state.

`Active` In the *Active* state, the local router is trying to initiate a TCP session with its peer. If the session establishes successfully, an Open message is sent and the local router transitions to the `OpenSent` state.

If the TCP session fails to establish, the local router initiates another session, sets the ConnectRetry timer to 0, and transitions back to the `Connect` state.

Attempts by the remote router to connect from an unexpected IP address for the session causes the local router to refuse the connection. The local router remains in the `Active` state and resets the ConnectRetry timer.

`OpenSent` The *OpenSent* state is reached upon a successful TCP establishment. The local router sends a BGP Open message and waits for an Open message from the remote peer.

When a valid Open message is received, the local router begins to send Keepalive messages to the remote router. The BGP peers negotiate the session parameters and the local router transitions to the `OpenConfirm` state.

Should a TCP disconnect be received while in this state, the local router terminates the BGP session, resets the ConnectRetry timer, and transitions back to the `Active` state.

`OpenConfirm` When the local router receives a valid Open message from the remote peer, the *OpenConfirm* state is reached. The local router sends Keepalive messages to the peer and waits for a Keepalive message in return.

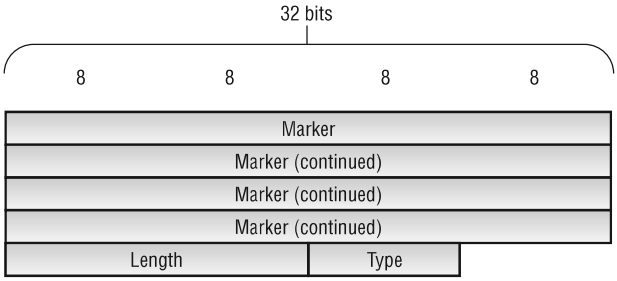
`Established` The *Established* state is achieved when a Keepalive message is received while in the `OpenConfirm` state. This is the final state of a peer relationship and designates a fully operational connection.

Two BGP peers can exchange routing information only when the `Established` state is reached. All other BGP peering states designate a nonfunctional session.

Message Types

In the previous section, we discussed some BGP message types that were exchanged between two peers. In total, BGP defines four messages types: Open, Update, Notification, and Keepalive. Each message contains a 19-octet fixed-size header, as seen in Figure 8.6.

FIGURE 8.6 BGP message header



The header consists of the following fields:

Marker (16 octets) This field is set to all 1s to detect a loss of synchronization. An Open message with authentication configured contains the authentication data for the session.

Length (2 octets) The total length of the BGP message is encoded in this field. Possible values range from 19 to 4096.

Type (1 octet) The type of BGP message is located in this field. Four type codes have been defined:

- 1 for an Open message
- 2 for an Update message
- 3 for a Notification message
- 4 for a Keepalive message

Let's now examine each of the BGP message types in more detail.

The Open Message

The *Open message* is the first packet BGP sends to a peer after the TCP connection has been established. It allows the two peers to negotiate the parameters of the peer session. These parameters include the BGP version, the hold time for the session, authentication data, refresh capabilities, and support for multiple *Network Layer Reachability Information (NLRI)*.

The message format is shown in Figure 8.7. After the common BGP header, the remaining fields include the following:

Version (1 octet) This field contains the current version of the peer. The default value is 4 and is set automatically.

Local Autonomous System (2 octets) The sender's AS value is encoded in this field.

Hold Time (2 octets) The sender places its proposed hold-time value here, and the two peers negotiate this value to the lower of the two proposals. The default value for the JUNOS software is 90 seconds, with a possible range between 6 and 65,535 seconds.

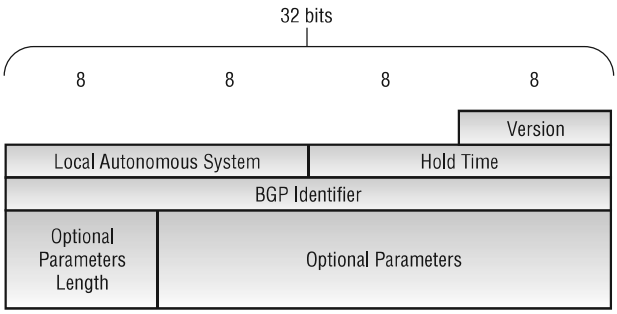
Each peer divides the negotiated hold time by 3 to calculate the Keepalive timer for the session.

BGP Identifier (4 octets) The local Router ID of the peer is encoded in this field. This uniquely identifies the router to the network.

Optional Parameters Length (1 octet) This field specifies the total length of the Optional Parameters field. A value of 0 indicates that no parameters are included in the message.

Optional Parameters (Variable) This variable-length field encodes any optional parameters used by the local peer. Possibilities include support for route refresh, authentication, and various NLRI. Each parameter is encoded in a (Type, Length, Value) triple. The parameter's type and length fields are 1 octet each and are followed by a variable-length value field.

FIGURE 8.7 The BGP Open message



The Update Message

Routing information is sent and withdrawn in BGP using the *Update message*. If needed, each message contains information previously advertised by the local router that is no longer valid. The same message may also contain new information advertised to the remote peer.

Each Update contains a single set of BGP attributes and all routes using those attributes. This format reduces the total number of packets routers send between BGP peers when exchanging routing knowledge.

The various attributes used by BGP are discussed in the "BGP Attributes" section later in this chapter.

Figure 8.8 shows the format of the Update message. The following fields follow the common BGP header:

Unfeasible Routes Length (2 octets) This field specifies the length of the Withdrawn Routes field that follows. A value of 0 designates that no routes are being withdrawn with this Update.

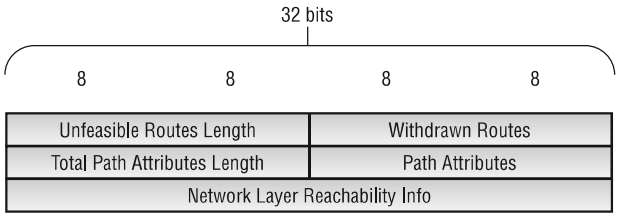
Withdrawn Routes (Variable) This field lists the routes previously announced that are now being withdrawn. Each route is encoded as a (Length, Prefix) tuple where the length is the number of bits in the subnet mask and the prefix is the IPv4 NLRI.

Total Path Attributes Length (2 octets) This field specifies the length of the Path Attributes field that follows. A value of 0 designates that no routes are being advertised with this message.

Path Attributes (Variable) The attributes of the path advertisement are contained in this field. Each attribute is encoded as a (Type, Length, Value) triple.

Network Layer Reachability Information (Variable) This field lists the routes advertised to the remote peer. Each route is encoded as a (Length, Prefix) tuple where the length is the number of bits in the subnet mask and the prefix is the IPv4 route.

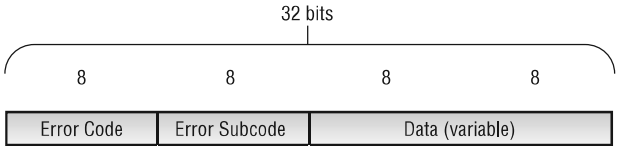
FIGURE 8.8 The BGP Update message



The Notification Message

When a BGP peer detects an error within the session, it sends a *Notification message* to the remote router and immediately closes both the BGP and TCP sessions. The format of the message is shown in Figure 8.9.

FIGURE 8.9 The BGP Notification message



In addition to the common BGP header, the Notification message contains the following:

Error Code (1 octet) This field specifies the type of BGP error seen by the local router. Six error codes have been defined:

- 1 for a Message Header Error
- 2 for an Open Message Error
- 3 for an Update Message Error
- 4 for Hold Time Expired
- 5 for a Finite State Machine Error
- 6 for a Cease

Error Subcode (1 octet) This field contains more specific information about the error. For example, within the Open Message Error type code, Authentication Failure is a possible subcode.

Data (Variable) This field is used to assist the administrator in troubleshooting the error. The contents of the field depend on the specific error code and subcode.

One common reason for a Notification message is intervention by a user. When you manually clear a BGP connection, you generate a Cease (Error Code 6) Notification message on the local router. The BGP and TCP sessions are torn down and reestablished. In addition, a configuration change that alters the parameters of an existing session will generate a Cease message by the local router.

The Keepalive Message

A BGP *Keepalive message* contains only the 19-octet message header and no other data. These messages are exchanged at one-third the negotiated hold-time value for the session, if necessary. The advertisement of an Update message within the keepalive period resets the timer to 0. In short, a Keepalive is sent only in the absence of other messages for a particular session.

Should the local router not receive a Keepalive or Update message within the hold-time period, a Notification message of *Hold Time Expired* is generated and the session is torn down.

Routing Information Bases

At this point, we've explored how a BGP router forms peer sessions with both external and internal peers and what routes are advertised to each peer. We've also looked at the message types used to actually form those peer relationships and advertise routes. We now need to explore the internal processes of how the local router receives, selects, and advertises specific routing information.

Each BGP router establishes memory locations in which to store routing knowledge. These are collectively known as a *Routing Information Base (RIB)*. A BGP peer maintains three categories of RIBs: the Adjacency-RIB-In, the Local-RIB, and the Adjacency-RIB-Out. Let's discuss each of these in further detail.

Adjacency-RIB-In

An *Adjacency-RIB-In* table is created on the local router for each established BGP peer. All routes received from the peer are placed in the appropriate memory table. There's one notable exception to this rule: Routes containing an AS Path loop are immediately discarded by the local router.

After receiving an Update message, the local router implements any applied import routing policies. These policies may alter the values of existing attributes, add new attributes, or discard specific routes. The router then examines all Adjacency-RIB-In tables for the best path advertisement to each unique destination. We discuss the specific method of selecting the best path in the "Route Selection Process" section later in this chapter.

Local-RIB

The best path to each destination is stored in the *Local-RIB* table. These are the routes that the local router uses to forward user data traffic. Only a single path advertisement per destination is placed in this table.

Adjacency-RIB-Out

Each established BGP peer also creates its own *Adjacency-RIB-Out* table for outbound route advertisements. Only routes currently located in the Local-RIB are eligible to be placed in this outbound database. In other words, a BGP router advertises only routes that it is currently using to forward data traffic.

By default, all Local-RIB routes are placed in each Adjacency-RIB-Out table. You can alter this behavior by applying export routing policies. A policy may add, alter, or remove attributes from each route. In addition, a policy may suppress a route from being advertised to a specific peer.

The Route Selection Process

As the local router scans the Adjacency-RIB-In tables for routing knowledge, it parses the feasible routes through a selection algorithm to determine the best path to each destination. The main criteria for this selection algorithm are the various BGP attributes, which we discuss in the next section, "BGP Attributes." Each vendor uses a slightly different algorithm, because the specifications dictate only the attributes to use with their possible values. We'll focus on how a Juniper Networks router makes its decisions.

If only a single version of a route exists in the inbound databases, that route is placed in the Local-RIB table and used for data forwarding. When multiple routes exist, they are grouped to evaluate their attributes. Each step in the selection algorithm attempts to eliminate all but one of the feasible routes to a destination. If multiple routes still exist after a particular step, the next step is executed. In this manner, the algorithm runs for only as long as it needs to.

The steps of the BGP selection algorithm are as follows:

1. The Next Hop attribute value for each route must be reachable in the local routing table; otherwise, the local router discards the route.
2. The router selects the route with the highest Local Preference attribute value.
3. The router selects the route with the shortest AS Path length.
4. The router selects the route with the smallest Origin attribute value.
5. The router selects the route with the smallest Multiple Exit Discriminator attribute value. This step is executed, by default, only for routes from the same neighboring AS.
6. The router selects routes learned from an EBGP peer over routes learned from an IBGP peer. If the remaining routes are all EBGP-learned routes, the router skips to step 9.
7. The router selects the route with the smallest IGP metric to the advertised BGP Next Hop.
8. If Route Reflection is used for IBGP peering, the router selects the route with the shortest Cluster-List length.

9. The router selects the route from the peer with the smallest numerical Router ID.
10. The router selects the route from the peer with the smallest numerical Peer Address.

BGP Attributes

The selection of a BGP route for data forwarding is highly dependent on the value of the path attributes. Each attribute is encoded in an Update message using a TLV triple. The Type portion of the TLV is a 2-octet field representing the following:

Optional Bit (Bit 0) An attribute is either well known (a value of 0) or optional (a value of 1).

Transitive Bit (Bit 1) Optional attributes can be either nontransitive (a value of 0) or transitive (a value of 1). Well-known attributes are always transitive.

Partial Bit (Bit 2) Only optional transitive attributes use this bit. A 0 means each BGP router along the path recognized this attribute. A 1 means that at least one BGP router along the path did not recognize the attribute.

Extended Length Bit (Bit 3) This bit sets the size of the TLV Length portion to one octet (a value of 0) or two octets (a value of 1).

Unused (Bits 4–7) These bit positions are not used and must be set to 0.

Type Code (Bits 8–15) The specific kind of attribute is encoded in this 1-octet field.

The attribute type bits result in four main categories of BGP attributes: well-known mandatory, well-known discretionary, optional transitive, and optional nontransitive.

BGP routers must recognize all of the well-known attributes, which must be included with all routes. Discretionary attributes may or may not be included on a particular route. BGP routers do not have to understand optional attributes but must readvertise them based on their transitive setting. Transitive attributes are advertised to all BGP peers, while nontransitive attributes may be discarded if the local router doesn't recognize them.

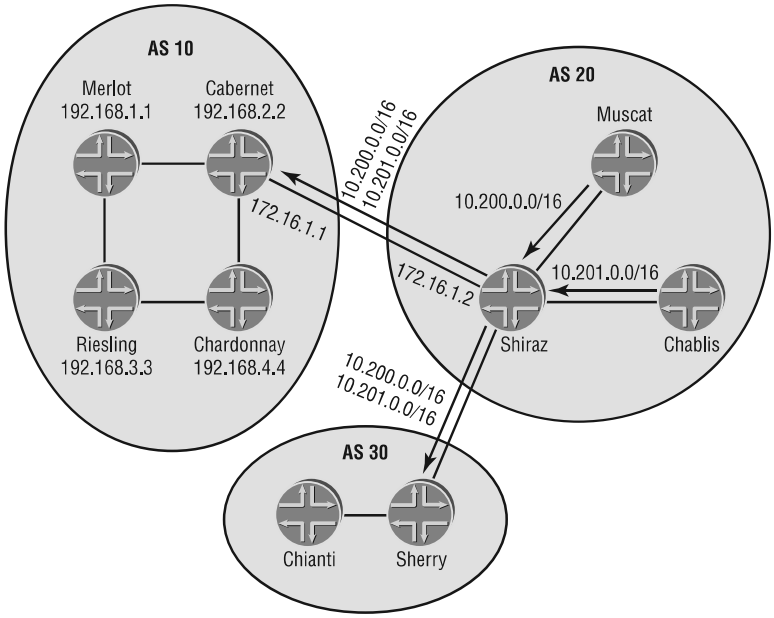
We now discuss some of the BGP attributes in further detail.

Next Hop

Next Hop, attribute type code 3, is a well-known mandatory attribute. It supplies each BGP router with the IP address of the next hop to the route destination. The local router performs a recursive lookup in the routing table to locate a route to the BGP next hop. The result of this recursive lookup is the physical next hop assigned to the BGP route in the routing and forwarding tables.

Reachability to the Next Hop attribute is critical to the operation of BGP. Recall from the previous section "The Route Selection Process" that a route whose Next Hop is not known is unusable to the local router. Care should be taken to maintain reachability because this attribute, by default, is modified only when a route is advertised across an EBGP peer session. Figure 8.10 shows an EBGP route advertisement.

FIGURE 8.10 An EBGP Next Hop change

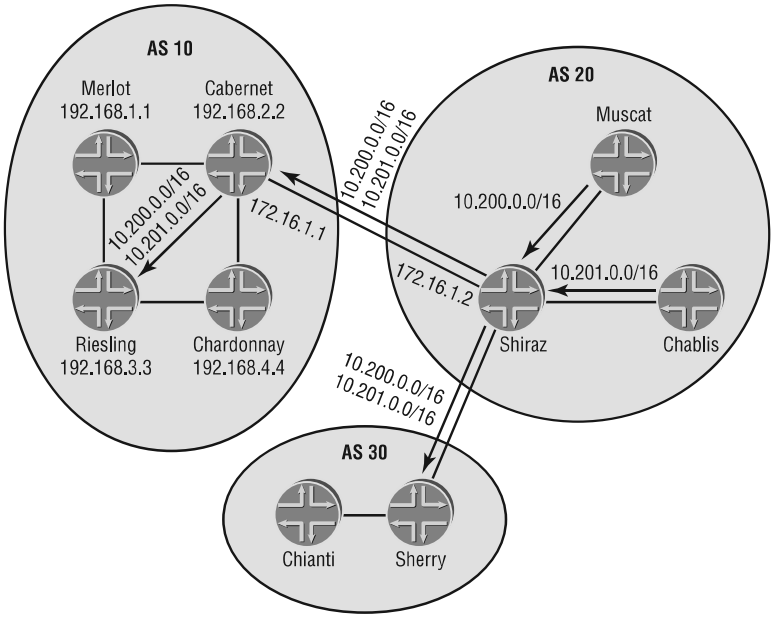


The Shiraz and Cabernet routers are EBGP peers in AS 10 and AS 20, respectively. As we discussed in the "Peers" section earlier in this chapter, they are directly connected and the BGP session is established between the 172.16.1.1 and 172.16.1.2 addresses. As Shiraz advertises the 10.200.0.0 /16 and 10.201.0.0 /16 routes to Cabernet, it changes the Next Hop attribute to the IP address on its side of the session—172.16.1.2 in our example. Cabernet is directly connected to the 172.16.1.0 /24 subnet, so reachability to the Next Hop is achieved.

Figure 8.11 shows an IBGP peer session between Cabernet and Riesling. When Cabernet advertises those same routes across that peer session, the Next Hop attribute is not changed—it is still 172.16.1.2. Chances are that Riesling does not have a route to the 172.16.1.0 /24 subnet because it is outside the AS boundary. This lack of reachability causes the 10.200.0.0 /16 and 10.201.0.0 /16 routes to become unusable to Riesling.

A Juniper Networks router marks these types of routes as hidden. You can view them by using the `show route hidden` command.

FIGURE 8.11 An IBGP Next Hop advertisement



To resolve this issue, the administrators of AS 10 must alter their current environment. There are five viable methods for providing reachability to the BGP Next Hop:

Setting the Next Hop attribute Considered by many to be a best practice for an IBGP peer session, this method allows the advertising router to change the value of the Next Hop attribute.

In Figure 8.11, Cabernet and Riesling are peers using their loopback addresses of 192.168.2.2 and 192.168.3.3. Cabernet changes the Next Hop to its peer address of 192.168.2.2 prior to advertising routes to Riesling. The BGP peer session was established using reachability from Riesling to 192.168.2.2. The Next Hop attribute inherently becomes reachable and the advertised routes are now usable.

Using an IGP passive interface In this scenario, the interface connecting the EBGP peers is configured to operate in passive mode within the IGP of the AS. This allows the IGP to advertise the external subnet as an internal route without establishing an adjacency between the EBGP peers. All routers in the AS now have reachability to the Next Hop advertised by the EBGP-speaking router.

Configuring an export policy The end result of this approach is exactly the same as operating the external interface in passive mode. The difference lies in the method of advertising the interface's subnet. In this instance, an export policy is configured within the IGP to advertise the `Direct` routes from the routing table. The external interface of the EBGP-speaking router now appears as an external IGP route on all routers in the AS and reachability to the Next Hop is achieved.

Establishing an IGP adjacency While a viable technical solution to solving Next Hop reachability, this is not a recommended practice for an ISP. This process involves establishing an IGP adjacency between the EBGP peers. The external subnet is automatically advertised to the IGP as part of this process and the Next Hop is reachable.

There is great risk involved in this process, however, because the IGPs are inherently trusting. All information advertised by the IGP from the neighboring AS is used without question and your internal routing tables may now contain routes you do not wish to use.

Using static routes This approach carries with it the advantages and disadvantages of using static routes in place of your IGP. Static routes are simple and easy to configure but require manual input into all routers in the AS. You must configure a separate static route for each Next Hop for which reachability is required.

Local Preference

Local Preference, attribute type code 5, is a well-known discretionary attribute. All BGP routers must understand Local Preference, but it is not required on every route. In fact, Local Preference is only used within the confines of an AS—no value is advertised to an EBGP peer.

Administrators use the Local Preference attribute to designate the exit point out of the AS. Two factors make the Local Preference attribute well suited for this task. First, every router within the AS has a Local Preference value assigned to all routes. Second, this attribute is the first tiebreaker used in the BGP route selection algorithm. Therefore, each BGP router in the network makes the same selection decision and all user data traffic flows to the router advertising the highest Local Preference value. By default, each route within an AS receives a Local Preference value of 100. Possible values range from 1 to 4,294,967,295.

AS Path

AS Path, attribute type code 2, is a well-known mandatory attribute. The attribute contains a sequenced list of AS numbers that represent the networks the route has transited.

This attribute is modified only when a route is advertised to an EBGP peer. During this process, the local router adds its AS number to the attribute. The new value is prepended to (added to the front of) the existing path attribute. This means that the AS Path is actually read in a right-to-left manner. Examining a route with the path `65001 65100 65250` tells us it was originally advertised to BGP by AS 65250. It was then advertised to a router in AS 65100, transited AS 65001, and finally arrived in the local AS 64699. When you advertise this route to an EBGP peer, you modify the path to become `64999 65001 65100 65250`.

In the "Network Connectivity" section earlier in this chapter, we discussed the AS Path attribute as a method for preventing routing loops. It is also used as a tiebreaker in the route selection algorithm. Each AS number represents a length of 1 with the shortest number of AS

hops in a path being preferred. For example, the AS Path `65000 65001` has a length of 2. It is shorter than the path `65100 65200 65300` but longer than the path `64800`.

In an attempt to influence traffic flows into your network, you can artificially lengthen the AS Path attribute. You do this when you advertise a route to an EBGP peer. Instead of adding your local AS number one time to the path, you add it several times. Assuming a local AS number of 64699, in our earlier example the default prepend action results in an AS Path length of 4 (`64999 65001 65100 65250`). Alternatively, you can prepend your AS number three times to result in an AS Path length of 6 (`64999 64999 64999 65001 65100 65250`).

Origin

Origin, attribute type code 1, is a well-known mandatory attribute. The Origin code designates the source of the route into BGP.

BGP routers receive and readvertise BGP routes by default, using the rules we outlined in the "Peers" section earlier in this chapter. However, the protocol does not naturally advertise non-BGP routes. Somewhere in the Internet, each route was explicitly configured for advertisement into BGP. For a Juniper Networks router, this is accomplished with an export policy, as we discussed in Chapter 4, "Routing Policy."

The first BGP router to advertise a route assigns a value to the Origin attribute to alert other routers as to the source of that route. The route selection algorithm may use this value as a tie-breaker, with a lower value being preferred. The current specification dictates three possible Origin values:

IGP The route was originally learned by an IGP on the source router. IGP is displayed with the character "I" and is encoded as a value of 0.

EGP The route was originally learned by the EGP protocol. EGP is displayed with the character "E" and is encoded as a value of 1.

Incomplete The route's source was unknown to the initial BGP router. Incomplete is displayed with the character "?" and is encoded as a value of 2.

The Origin letter code (`I/E/?`) is displayed at the end of the AS Path attribute. Therefore, an Origin of IGP appears as `65499 65000 I` in the router's output.

The JUNOS software always assigns an Origin code of IGP to all routes advertised using an export policy.

Multiple Exit Discriminator

Multiple Exit Discriminator (MED), attribute type code 4, is an optional nontransitive attribute. As such, BGP routes may not carry this attribute at all. Those that do, however, retain it only within the confines of a particular AS. This means that a MED attribute received from an EBGP peer is advertised to all IBGP peers and all routers may use the encoded value. MED values received from IBGP peers, however, are not readvertised to an EBGP peer. In other words, the router on the edge of the AS removes the attribute prior to sending the route.

By default, the MED attribute is only compared for routes received from the same neighboring AS. For example, suppose the route 192.168.100.0 /24 is received from AS 65000 and AS 65400. Any MED values associated with these routes are not comparable by the route selection algorithm since the first AS in the AS Path is not the same. On the other hand, should the 172.16.200.0 /24 route be received from two different routers in AS 64600, the MED values can be compared as a tiebreaker.

The JUNOS software interprets the absence of a MED as a value of 0.

Community

Community, attribute type code 8, is an optional transitive attribute. It is used to administratively group routes for a common policy action. For example, a BGP import policy might alter the Local Preference of all routes received with a certain community value assigned.

The attribute is encoded as a four-octet value where the first two octets represent an AS number and the remaining two octets represent a locally defined value. The Community attribute is displayed in the format `65001:1001`.

Real World Scenario

The BGP Attributes in Action

While it is nice to have a definition of the attributes and talk about their purpose, nothing drives home their use like an example. Let's reexamine the BGP route selection algorithm using a sample router.

Suppose a BGP router in AS 65001 peers with six different neighbors. Each neighbor advertises the 10.0.0.0 /8 route to the local router with various attributes defined. The received routes are:

- Peer A: Local Preference—100; AS Path—64777 64888; Origin—I; MED—10
- Peer B: Local Preference—200; AS Path—64777 64888; Origin—I; MED—5
- Peer C: Local Preference—200; AS Path—64777 64888 64999; Origin—I; MED—15
- Peer D: Local Preference—200; AS Path—64777 64888; Origin—I; MED—25
- Peer E: Local Preference—200; AS Path—64777 64888; Origin—?; MED—20
- Peer F: Local Preference—200; AS Path—64777 64888 64999 65000; Origin—I; MED—30

The local router examines the Adjacency-RIB-In tables and locates these versions of the 10.0.0.0 /8 route. The router then uses the BGP route selection algorithm to decide which version should be used for forwarding traffic and placed into the Local-RIB table.

Assuming that the BGP Next Hop for each route is reachable in the routing table, the local router first examines the Local Preference attribute. It finds that the route from Peer A has a value of 100, while every other version of the route has a value of 200. A higher Local Preference value is preferred over a lower value, so the router removes Peer A from its list of candidates.

The router then examines the AS Path length of the remaining routes. Peers B, D, and E advertised a path length of 2. Peer C's version of the route has a path length of 3, and Peer F advertised a path length of 4. A shorter AS Path length is preferred, and the router removes Peers C and F from the list of candidates.

The Origin attribute of the remaining routes from Peers B, D, and E is now evaluated. The local router finds the routes from Peers B and D to have an Origin of IGP (`I`), while the route from Peer E has an Origin of Incomplete (`?`). The IGP Origin is preferred over the Incomplete Origin, and the router removes Peer E from the candidate list.

The next attribute considered by the router is the Multiple Exit Discriminator. Before examining the value of the attribute, the router first must decide if the candidate routes were advertised from the same neighboring AS. Peer B's route was last in AS 64777, as was the route from Peer D. These routes meet the required criteria, and the router finds that the MED of Peer B's route is 5. This is lower than the MED value 25 on Peer D's route. A lower MED value is preferred, and the router removes Peer D from the candidate list.

With only a single route remaining to be evaluated, the local router has found the best path to 10.0.0.0/8. The route is placed into the Local-RIB table, and user data packets are forwarded to the BGP next hop advertised by Peer B.

Juniper Networks Implementation

We've discussed the basic theoretical aspects of using BGP on a Juniper Networks router. Next we examine the steps required to configure the protocol and advertise routes to various peers. There are multiple theories, methods, and commands you can use in your configuration. This provides you with the most flexibility in operating your router. We focus here only on basic steps and procedures.

The JUNOS software provides for configuration options at the global BGP level, at the peer-group level, or at a specific neighbor level. It is often a best practice to group your peers into peer groups for ease of administering routing policies. We follow this practice throughout the remainder of this chapter.

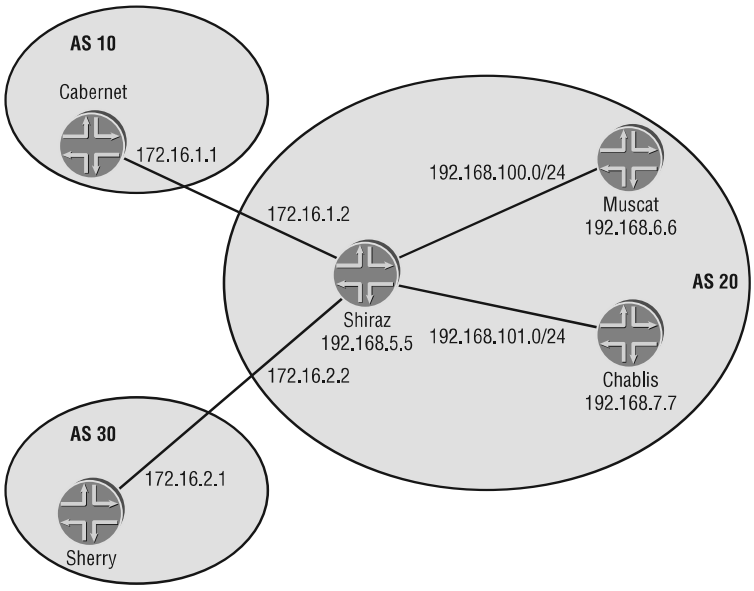
Establishing Peer Relationships

Before any routes are advertised or attributes modified, you must first establish your BGP peer sessions. In the "Peers" section earlier in this chapter, we discussed the differences between an EBGP and an IBGP peer session. As you might suspect, the configurations of these peer sessions are different as well. One common aspect to both, however, is the local AS number.

Assigning an AS Number

The first step toward configuring BGP is to tell the router its local AS number. You configure this value in the `[edit routing-options]` hierarchy. Figure 8.12 shows the Shiraz router within AS 20.

FIGURE 8.12 A BGP sample network



The `autonomous-system` command assigns the local AS number like so:

```
[edit]
user@Shiraz# set routing-options autonomous-system 20
```

The JUNOS software stores the AS number in the central `routing-options` hierarchy to allow for easier configuration. All BGP peers in the global routing instance inherit this value. In addition, other instances of BGP on the router (possibly within a VPN) use this common value.

Configuring an EBGP Peer Session

An external BGP peer session requires that the routers be directly connected to each other. Figure 8.12 shows the Cabernet router in AS 10 and the Sherry router in AS 30. Both routers are physically connected to Shiraz in AS 20.

A peer group called `ebgp-peers` has been created on Shiraz for the EBGP peer sessions to Cabernet and Sherry. The IP address and AS number of each peer is explicitly configured:

```
[edit protocols bgp group ebgp-peers]
user@Shiraz# set neighbor 172.16.1.1 peer-as 10
user@Shiraz# set neighbor 172.16.2.1 peer-as 30
```

We also inform the router whether the peer-group members are EBGP peers or IBGP peers by using the `type` command. This allows the group to perform the default attribute changes if required.

```
[edit protocols bgp group ebgp-peers]
user@Shiraz# set type external
```

The resulting configuration looks like this:

```
[edit protocols bgp]
user@Shiraz# show
group ebgp-peers {
    type external;
    neighbor 172.16.1.1 {
        peer-as 10;
    }
    neighbor 172.16.2.1 {
        peer-as 30;
    }
}
```

The configurations for Cabernet and Sherry mirror the peer-group setup and explicit configuration found on Shiraz:

```
[edit]
user@Cabernet# show
routing-options {
    autonomous-system 10;
}
protocols {
    bgp {
        group AS-20 {
            type external;
```

```
            peer-as 20;
            neighbor 172.16.1.2;
        }
    }
}

[edit]
user@Sherry# show
routing-options {
    autonomous-system 30;
}
protocols {
    bgp {
        group AS-20 {
            type external;
            peer-as 20;
            neighbor 172.16.2.2;
        }
    }
}
```

Configuring an IBGP Peer Session

In the "Peers" section earlier in this chapter, we explained that IBGP peers require only IP reachability to form a peer session. Often that reachability is achieved through an IGP within the AS. Using Figure 8.12 as a guide, we see that AS 20 consists of the Shiraz, Muscat, and Chablis routers. OSPF is configured within the domain, and Shiraz has IGP routes to the other routers:

```
user@Shiraz> show route protocol ospf

inet.0: 14 destinations, 15 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.6.6/32     *[OSPF/10] 00:00:06, metric 1
                    > to 192.168.100.2 via ge-0/2/0.0
192.168.7.7/32     *[OSPF/10] 00:00:06, metric 1
                    > via at-0/1/0.100
```

A new peer group called `ibgp-peers` has been created on Shiraz. The IP addresses of Muscat and Chablis are configured and the peer group is designated as an internal group by using the `type` command:

```
[edit protocols bgp group ibgp-peers]
```

```
user@Shiraz# set type internal
user@Shiraz# set neighbor 192.168.6.6
user@Shiraz# set neighbor 192.168.7.7
```

The loopback addresses of the IBGP peers are used to provide redundancy and resiliency to the internal BGP network. Should a physical interface address be configured, its failure would cause the peer session to drop. In our case, as long as the router is reachable through any means, the BGP session remains established. The use of loopback addresses for peer sessions also requires an additional configuration step on Shiraz. We inform the BGP process of our own loopback address by using the `local-address` command:

```
[edit protocols bgp group ibgp-peers]
user@Shiraz# set local-address 192.168.5.5
```

The configuration on Shiraz now looks like this:

```
[edit protocols bgp]
user@Shiraz# show
group ebgp-peers {
    type external;
    neighbor 172.16.1.1 {
        peer-as 10;
    }
    neighbor 172.16.2.1 {
        peer-as 30;
    }
}
group ibgp-peers {
    type internal;
    local-address 192.168.5.5;
    neighbor 192.168.6.6;
    neighbor 192.168.7.7;
}
```

Both the Muscat and Chablis routers have a similar `ibgp-peers` configuration to establish an IBGP full-mesh within AS 20.

We didn't configure an AS number for the IBGP peers because we used the `type internal` command. This command informs the router that the local AS number in `routing-options` should be used for these peer sessions.

Real World Scenario

Using *local-address*

It might seem strange to you that configuring an IBGP session requires more commands than an EBGP session does. After all, these peers are inside your own AS, and you trust their services, so it should be easy to set up. The truth is that it has nothing to do with trust, but rather with the method that BGP operates.

Each peer session requires that the peer's IP address and AS number be explicitly configured in order for the session to become established. This information is exchanged in Open messages during session establishment and must be agreed on for the session to come up. The setting of AS numbers is straightforward in the configuration. It is the IP address of the peer that causes a possible issue.

One basic principle of IP packets is that each contains a source and destination IP address. When a router generates an IP packet itself, the source IP address becomes the address of the outbound interface. For EBGP peers, this default behavior is fine because the peers are connected and are peering across that interface address. IBGP peers, on the other hand, are peering to the loopback address of the remote router. Let's look at a small example.

Router A and Router B want to become IBGP peers using their loopback addresses of 1.1.1.1 and 2.2.2.2, respectively. When Router A sends its Open message to 2.2.2.2, the source IP address of the packet is the outgoing interface, say 10.10.10.1. Router B receives this Open message and compares the source address of the packet to its list of configured peers. Router B does not find a configuration for 10.10.10.1 and rejects the Open message. Router B performs this same process in reverse, with Router A rejecting the Open message from Router B. Two routers in this situation remain in the BGP `Active` state forever, a clear indication of this problem.

The resolution to this issue is the `local-address` command. Its function is quite simple—it changes the source IP address of the BGP messages. In our example, Router A configures `local-address 1.1.1.1` within its peer session to Router B. The BGP Open message now lists 1.1.1.1 as the source IP address of the packet. Router B receives the message and compares the source address to its list of configured peers. It now finds a match for Router A and responds to the Open message with a BGP Keepalive message. The two peers can now become established and advertise routing knowledge to each other.

Verifying Your BGP Sessions

Now that the configuration of our peers is complete, we would like to verify that the peer sessions are actually established and operational. There are three main commands available in the JUNOS software to accomplish this. Each provides different levels of information to the user. Let's examine them in greater detail using Figure 8.12 as a guide.

show bgp summary

The `show bgp summary` command provides you with a good snapshot of the protocol on your router. This is often the command you use to determine if your peer sessions are established. The Shiraz router has configured two EBGP and two IBGP peers. The output below reveals the state of each peer:

```
user@Shiraz> show bgp summary
Groups: 2 Peers: 4 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                12         12          0          0          0          0
Peer                AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State
172.16.1.1          10        428        430       0       0     3:33:00 4/4/0
172.16.2.1          30        428        430       0       0     3:32:56 4/4/0
192.168.6.6         20        392        392       0       0     3:14:30 2/2/0
192.168.7.7         20        390        391       0       0     3:14:02 2/2/0
```

The command output is separated into a summary section and a detailed section for the specific peers. Within the summary area, you can see the number of peer groups and peers configured. You also can determine some routing information for received BGP routes. In the case of Shiraz, 12 path advertisements have been received from peers, and all 12 are currently active and being used to forward user traffic.

The specific peer portion of the `show bgp summary` output contains the following fields:

`Peer` The configured peer address for each peer is listed in this field.

`AS` The configured AS number for each peer is listed here.

`InPkt` This field lists the total number of BGP messages received from the peer.

`OutPkt` This field lists the total number of BGP messages sent to each peer.

`OutQ` This field displays the number of BGP messages queued and waiting to be sent to a peer. This number is usually 0 because the queue is emptied quickly. A high and constant value in this column may indicate a problem with the peer session.

`Flaps` This field specifies the number of times the peer session has been closed and reestablished.

`Last Up/Dwn` This field displays the amount of time since the peer last changed state from `Established` to any other BGP state, or to Established from any other BGP state.

`State` This field shows the current BGP state of the peer. When the session reaches the `Established` state, routing information is displayed instead of a keyword. The routes are shown in the format `<# Active>/<# Received</<# Damped>`. For example, Shiraz has received four routes from its peer at 172.16.2.1. All four routes are currently active in the routing table and none of them have been damped.

show bgp group

To view the configured peer groups on your router, use the `show bgp group` command. This displays each group's name, type, and configured peers. The output from Shiraz shows:

```
user@Shiraz> show bgp group
Group Type: External                    Local AS: 20
  Name: ebgp-peers
  Total peers: 2        Established: 2
  172.16.1.1+179
  172.16.2.1+179
  Route Queue Timer: unset Route Queue: empty

Group Type: Internal    AS: 20          Local AS: 20
  Name: ibgp-peers
  Total peers: 2        Established: 2
  192.168.6.6+1910
  192.168.7.7+1127
  Route Queue Timer: unset Route Queue: empty
```

The IP addresses displayed are the configured BGP peer addresses. The additional information details the TCP port number used for the session. For example, Shiraz is peering with 172.16.2.1 and the remote port number is 179. This tells you that Shiraz established the peer session because it is using the well-known port for BGP. The situation is a bit different for the 192.168.6.6 peer, whose remote TCP port is 1910. The far-end router initiated the session to Shiraz using the well-known BGP port number.

show bgp neighbor

To receive the most detailed information about your BGP peers, use the `show bgp neighbor` command. The output from Shiraz for just the 172.16.1.1 peer is as follows:

```
user@Shiraz> show bgp neighbor 172.16.1.1
Peer: 172.16.1.1+179  AS 10    Local: 172.16.1.2+1028 AS 20
  Type: External    State: Established    Flags: <>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference HoldTime PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.2.2      Local ID: 192.168.5.5      Active Holdtime: 90
  Keepalive Interval: 30
  Local Interface: so-0/0/1.0
  NLRI advertised by peer: inet-unicast
```

```
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Table inet.0 Bit: 10000
    Send state: in sync
    Active prefixes: 4
    Received prefixes: 4
    Suppressed due to damping: 0
  Last traffic (seconds): Received 13   Sent 13   Checked 13
  Input messages:  Total 438    Updates 4       Refreshes 0     Octets 8473
  Output messages: Total 440    Updates 4       Refreshes 0     Octets 8526
  Output Queue[0]: 0
```

The output of this command contains a wealth of information, including the following:

- The peer address and AS number of the peer
- Configured hold time (`Holdtime`) and negotiated hold time (`Active Holdtime`)
- Router ID values for each peer
- The number of active and received routes
- The number of packets and the amount of octets sent to and received from the peer

Viewing Routing Knowledge

BGP routers by default advertise only active BGP routes in the routing table. This creates a sort of chicken-and-egg problem. A route can appear in the routing table as a BGP route only if it is received from a BGP peer, but a BGP peer can only advertise a route if it's already in the routing table as a BGP route.

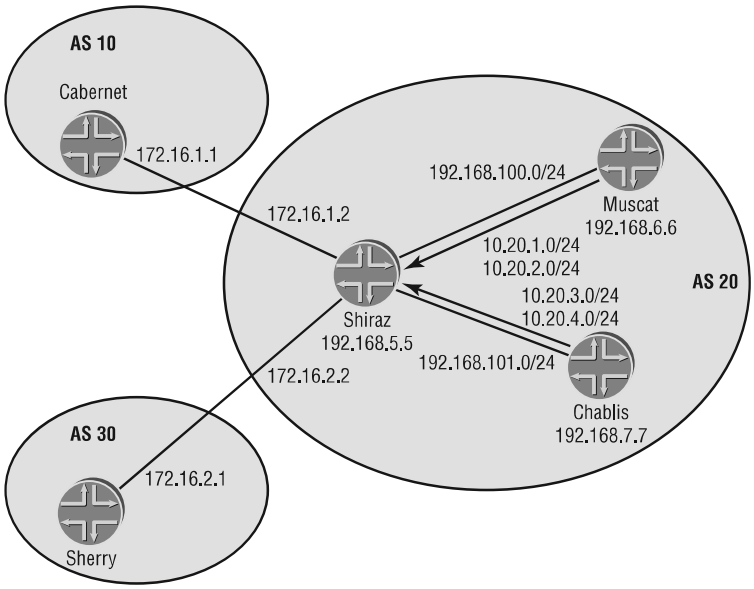
Injecting Routes into BGP

The solution to this problem is the explicit advertisement of other routes into BGP. For a Juniper Networks router, this is accomplished with an `export` routing policy.

Figure 8.13 shows some BGP routes advertised into AS 20 by Muscat and Chablis. Each router has local static routes representing customer networks that are injected with a routing policy called ***send-statics***. The policy configuration on Muscat is:

```
[edit]
user@Muscat# show policy-options
policy-statement send-statics {
    term find-static-routes {
        from protocol static;
        then accept;
    }
}
```

FIGURE 8.13 BGP routing knowledge



The policy is applied to BGP using the `export` command:

```
[edit protocols bgp group ibgp-peers]
user@Muscat# set export send-statics
```

The resulting configuration looks like this:

```
[edit protocols bgp]
user@Muscat# show
group ibgp-peers {
    type internal;
    local-address 192.168.6.6;
    export send-statics;
    neighbor 192.168.5.5;
    neighbor 192.168.7.7;
}
```

Viewing Received Routes

The routes that are received from each established BGP peer are stored in the Adjacency-RIB-IN tables. Within the JUNOS software, this is actually part of the routing table. Database pointers are

used to keep the routing knowledge separate. You can view the routes advertised by a peer on the local router by using the `show route receive-protocol bgp` ***`neighbor-address`*** command.

The routes advertised from Chablis, whose peer address is 192.168.7.7, are as follows:

```
user@Shiraz> show route receive-protocol bgp 192.168.7.7

inet.0: 26 destinations, 27 routes (26 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.20.3.0/24
192.168.7.7                 0        100 I
10.20.4.0/24
192.168.7.7                 0        100 I
```

Both the 10.20.3.0 /24 and 10.20.4.0 /24 routes are being received. The BGP Next Hop is 192.168.7.7 for both routes. Other BGP attributes visible in the output include a MED of 0, a Local Preference of 100, and an Origin of IGP.

Viewing Advertised Routes

The same routes are visible from the perspective of Chablis. They are also stored in the routing table with database pointers representing the Adjacency-RIB-Out. You can view this information by using the `show route advertising-protocol bgp` ***`neighbor-address`*** command:

```
user@Chablis> show route advertising-protocol bgp 192.168.5.5

inet.0: 21 destinations, 22 routes (13 active, 0 holddown, 8 hidden)
+ = Active Route, - = Last Active, * = Both

10.20.3.0/24
Self                        0        100 I
10.20.4.0/24
Self                        0        100 I
```

The main difference in this output is the listing of the BGP Next Hop attribute as `Self`. This represents the peer address of the router, 192.168.7.7 in our case. It is common to view this representation when using this command.

Viewing Local Routes

The representation of the BGP Local-RIB database is the routing table on your router. You can view this information by using a simple `show route` command. Of course, this shows you all the routes known to your router. To view just the BGP-learned routes, add the *`protocol bgp`* option.

The BGP routes on Shiraz are:

```
user@Shiraz> show route protocol bgp

inet.0: 26 destinations, 27 routes (26 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.1.0/24       *[BGP/170] 00:23:21, MED 0, localpref 100
                      AS path: 10 I
                    > to 172.16.1.1 via so-0/0/1.0
10.10.2.0/24       *[BGP/170] 00:23:21, MED 0, localpref 100
                      AS path: 10 I
                    > to 172.16.1.1 via so-0/0/1.0
10.10.3.0/24       *[BGP/170] 00:23:21, MED 0, localpref 100
                      AS path: 10 I
                    > to 172.16.1.1 via so-0/0/1.0
10.10.4.0/24       *[BGP/170] 00:23:21, MED 0, localpref 100
                      AS path: 10 I
                    > to 172.16.1.1 via so-0/0/1.0
10.20.1.0/24       *[BGP/170] 02:37:11, MED 0, localpref 100, from 192.168.6.6
                      AS path: I
                    > to 192.168.100.2 via ge-0/2/0.0
10.20.2.0/24       *[BGP/170] 02:37:11, MED 0, localpref 100, from 192.168.6.6
                      AS path: I
                    > to 192.168.100.2 via ge-0/2/0.0
10.20.3.0/24       *[BGP/170] 02:36:34, MED 0, localpref 100, from 192.168.7.7
                      AS path: I
                    > via at-0/1/0.100
10.20.4.0/24       *[BGP/170] 02:36:34, MED 0, localpref 100, from 192.168.7.7
                      AS path: I
                    > via at-0/1/0.100
10.30.1.0/24       *[BGP/170] 00:24:18, MED 0, localpref 100
                      AS path: 30 I
                    > to 172.16.2.1 via so-0/0/0.0
10.30.2.0/24       *[BGP/170] 00:24:18, MED 0, localpref 100
                      AS path: 30 I
                    > to 172.16.2.1 via so-0/0/0.0
```

```
10.30.3.0/24       *[BGP/170] 00:24:18, MED 0, localpref 100
                      AS path: 30 I
                    > to 172.16.2.1 via so-0/0/0.0
10.30.4.0/24       *[BGP/170] 00:24:18, MED 0, localpref 100
                      AS path: 30 I
                    > to 172.16.2.1 via so-0/0/0.0
```

All of the BGP attributes are visible when you use the `show route detail` command. We'll examine just the 10.20.3.0 /24 route advertised from Chablis:

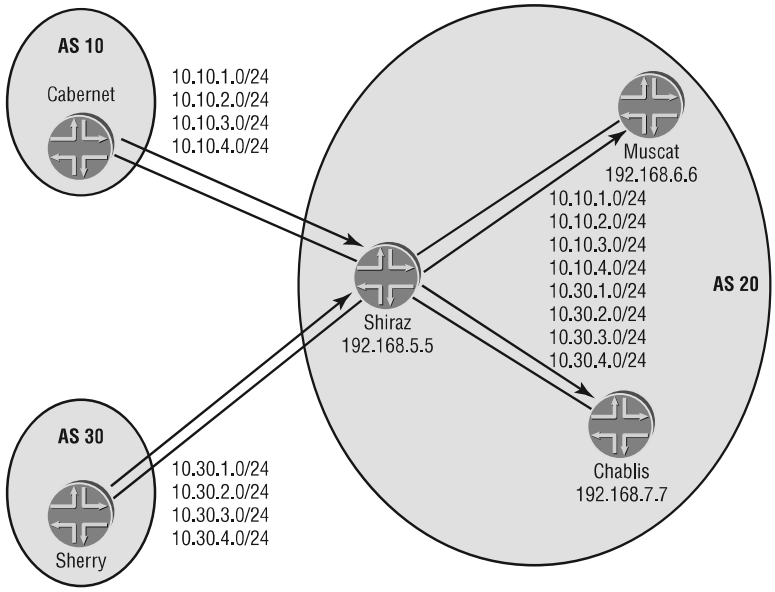
```
user@Shiraz> show route 10.20.3/24 detail

inet.0: 26 destinations, 27 routes (26 active, 0 holddown, 0 hidden)
10.20.3.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Source: 192.168.7.7
                Nexthop: via at-0/1/0.100, selected
                Protocol Nexthop: 192.168.7.7 Indirect nexthop: 8458088 44
                State: <Active Int Ext>
                Local AS:    20 Peer AS:    20
                Age: 2:39:44    Metric: 0       Metric2: 1
                Task: BGP_20.192.168.7.7+1127
                Announcement bits (3): 0-KRT 3-BGP.0.0.0.0+179 4-Resolve inet.0
                AS path: I
                Localpref: 100
                Router ID: 192.168.7.7
```

The two attributes not explicitly listed by name are MED and Next Hop. The MED value is encoded within the `Metric` output and is currently set to 0. The `Metric2` field displays the current IGP cost to the BGP Next Hop and is set to 1. The Next Hop is listed as the `Protocol Nexthop` output and is 192.168.7.7, the peer address of the IBGP peer. This notation differentiates the attribute from the physical forwarding next hop (`at-0/1/0.100`) displayed in the `Nexthop` field.

Solving Next Hop Reachability

During our earlier discussion of the BGP attributes, we found that possible issues existed with establishing reachability to the BGP Next Hop attribute. This is especially noticeable for IBGP peer sessions since the attribute is changed, by default, only across EBGP sessions. Figure 8.14 shows that Cabernet in AS 10 and Sherry in AS 30 are advertising routes to Shiraz in AS 20.

FIGURE 8.14 Next Hop reachability



As shown here, the routes are visible in the local routing table of Shiraz:

```
user@Shiraz> show route protocol bgp terse

inet.0: 26 destinations, 27 routes (26 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A Destination        P Prf   Metric 1   Metric 2  Next hop        AS path
* 10.10.1.0/24       B 170        100          0 >172.16.1.1      10 I
* 10.10.2.0/24       B 170        100          0 >172.16.1.1      10 I
* 10.10.3.0/24       B 170        100          0 >172.16.1.1      10 I
* 10.10.4.0/24       B 170        100          0 >172.16.1.1      10 I
* 10.20.1.0/24       B 170        100          0 >192.168.100.2   I
* 10.20.2.0/24       B 170        100          0 >192.168.100.2   I
* 10.20.3.0/24       B 170        100          0 >at-0/1/0.100    I
* 10.20.4.0/24       B 170        100          0 >at-0/1/0.100    I
* 10.30.1.0/24       B 170        100          0 >172.16.2.1      30 I
* 10.30.2.0/24       B 170        100          0 >172.16.2.1      30 I
* 10.30.3.0/24       B 170        100          0 >172.16.2.1      30 I
* 10.30.4.0/24       B 170        100          0 >172.16.2.1      30 I
```

A detailed examination of a route from Cabernet and Sherry reveals that the BGP Next Hop is currently set to the addresses 172.16.1.1 and 172.16.2.1, respectively:

```
user@Shiraz> show route 10.10.1/24 detail

inet.0: 26 destinations, 27 routes (26 active, 0 holddown, 0 hidden)
10.10.1.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Source: 172.16.1.1
                Nexthop: 172.16.1.1 via so-0/0/1.0, selected
                State: <Active Ext>
                Local AS:    20 Peer AS:    10
                Age: 57:35      Metric: 0
                Task: BGP_10.172.16.1.1+179
                Announcement bits (3): 0-KRT 3-BGP.0.0.0.0+179 4-Resolve inet.0
                AS path: 10 I
                Localpref: 100
                Router ID: 192.168.2.2

lab@Shiraz> show route 10.30.1/24 detail

inet.0: 26 destinations, 27 routes (26 active, 0 holddown, 0 hidden)
10.30.1.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Source: 172.16.2.1
                Nexthop: 172.16.2.1 via so-0/0/0.0, selected
                State: <Active Ext>
                Local AS:    20 Peer AS:    30
                Age: 58:40      Metric: 0
                Task: BGP_30.172.16.2.1+179
                Announcement bits (3): 0-KRT 3-BGP.0.0.0.0+179 4-Resolve inet.0
                AS path: 30 I
                Localpref: 100
                Router ID: 192.168.8.8
```

Following the rules outlined in the "Peers" section earlier in this chapter, Shiraz advertises all active EBGP-learned routes to both its IBGP peers. The specific routes advertised to Chablis include:

```
user@Shiraz> show route advertising-protocol bgp 192.168.7.7

inet.0: 26 destinations, 27 routes (26 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.10.1.0/24
172.16.1.1               0        100 10 I
10.10.2.0/24
172.16.1.1               0        100 10 I
10.10.3.0/24
172.16.1.1               0        100 10 I
10.10.4.0/24
172.16.1.1               0        100 10 I
10.30.1.0/24
172.16.2.1               0        100 30 I
10.30.2.0/24
172.16.2.1               0        100 30 I
10.30.3.0/24
172.16.2.1               0        100 30 I
10.30.4.0/24
172.16.2.1               0        100 30 I
```

As per the BGP defaults, the Next Hop attribute was not changed as the routes were advertised to the IBGP peer. Let's see what routes Chablis actually received from Shiraz:

```
user@Chablis> show route receive-protocol bgp 192.168.5.5

inet.0: 21 destinations, 22 routes (13 active, 0 holddown, 8 hidden)

user@Chablis>
```

At first glance, it appears that Chablis did not receive any routes. A closer examination, however, reveals that some routes are currently marked as hidden. We can view these routes with the `show route hidden` command:

```
user@Chablis> show route hidden

inet.0: 21 destinations, 22 routes (13 active, 0 holddown, 8 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.1.0/24            [BGP/170] 01:04:41, MED 0, localpref 100, from 192.168.5.5
                          AS path: 10 I
                          Unusable
10.10.2.0/24            [BGP/170] 01:04:41, MED 0, localpref 100, from 192.168.5.5
                          AS path: 10 I
                          Unusable
10.10.3.0/24            [BGP/170] 01:04:41, MED 0, localpref 100, from 192.168.5.5
                          AS path: 10 I
                          Unusable
```

```
10.10.4.0/24        [BGP/170] 01:04:41, MED 0, localpref 100, from 192.168.5.5
                      AS path: 10 I
                      Unusable
10.30.1.0/24        [BGP/170] 01:05:38, MED 0, localpref 100, from 192.168.5.5
                      AS path: 30 I
                      Unusable
10.30.2.0/24        [BGP/170] 01:05:38, MED 0, localpref 100, from 192.168.5.5
                      AS path: 30 I
                      Unusable
10.30.3.0/24        [BGP/170] 01:05:38, MED 0, localpref 100, from 192.168.5.5
                      AS path: 30 I
                      Unusable
10.30.4.0/24        [BGP/170] 01:05:38, MED 0, localpref 100, from 192.168.5.5
                      AS path: 30 I
                      Unusable
```

It appears as though we've found our missing routes, but the routes are currently listed as `Unusable`. A closer look at the 10.10.1.0 /24 route reveals a vital clue:

```
user@Chablis> show route hidden 10.10.1/24 extensive

inet.0: 21 destinations, 22 routes (13 active, 0 holddown, 8 hidden)
10.10.1.0/24 (1 entry, 0 announced)
         BGP    Preference: 170/-101
                Next hop type: Unusable
                State: <Hidden Int Ext>
                Local AS:    20 Peer AS:    20
                Age: 1:07:23    Metric: 0
                Task: BGP_20.192.168.5.5+179
                AS path: 10 I
                Localpref: 100
                Router ID: 192.168.5.5
                Indirect nexthops: 1
                        Protocol Nexthop: 172.16.1.1 Indirect nexthop: 0 -
```

The BGP Next Hop of 172.16.1.1 is `Unusable`. It appears the recursive lookup did not find a route to the Next Hop. We can verify this with a `show route` command:

```
user@Chablis> show route 172.16.1.1

user@Chablis>
```

In the "BGP Attributes" section earlier in this chapter, we listed five viable methods for solving this problem. Let's follow the best practice recommendation and alter the Next Hop attribute on Shiraz as the routes are advertised to the IBGP peers. We accomplish this by applying an export routing policy to BGP. We first build the policy on Shiraz:

```
[edit]
user@Shiraz# show policy-options
policy-statement next-hop-self {
    term change-the-attribute {
        from protocol bgp;
        then {
            next-hop self;
        }
    }
}
```

The policy called ***next-hop-self*** matches all active BGP routes in the routing table. The action dictates that the Next Hop attribute be changed to the value `self`. In the "Viewing Advertised Routes" section earlier in this chapter, we explained that the keyword `self` translates into the local peer address for the BGP session. Shiraz is using its loopback address of 192.168.5.5 to peer with Chablis, so this address is used as the Next Hop value.

The attribute should be changed only as the routes are advertised to the peers and not for Shiraz itself. An export policy application within the IBGP peer group accomplishes this goal:

```
[edit protocols bgp]
user@Shiraz# set group ibgp-peers export next-hop-self

[edit protocols bgp]
user@Shiraz# show
group ebgp-peers {
    type external;
    neighbor 172.16.1.1 {
        peer-as 10;
    }
    neighbor 172.16.2.1 {
        peer-as 30;
    }
}
group ibgp-peers {
    type internal;
    local-address 192.168.5.5;
    export next-hop-self;
```

```
    neighbor 192.168.6.6;
    neighbor 192.168.7.7;
}
```

Do not apply a `next-hop-self` policy as an import policy for an EBGP peer. This results in all received routes being marked as `hidden` in the routing table, because a recursive lookup is not performed for EBGP peers.

A check of the routes advertised to Chablis reveals the Next Hop set to the value `self`:

```
user@Shiraz> show route advertising-protocol bgp 192.168.7.7

inet.0: 26 destinations, 27 routes (26 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.1.0/24
Self                    0        100 10 I
10.10.2.0/24
Self                    0        100 10 I
10.10.3.0/24
Self                    0        100 10 I
10.10.4.0/24
Self                    0        100 10 I
10.30.1.0/24
Self                    0        100 30 I
10.30.2.0/24
Self                    0        100 30 I
10.30.3.0/24
Self                    0        100 30 I
10.30.4.0/24
Self                    0        100 30 I
```

Chablis is now receiving the routes as:

```
user@Chablis> show route receive-protocol bgp 192.168.5.5

inet.0: 21 destinations, 22 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.1.0/24
192.168.5.5             0        100 10 I
```

```
10.10.2.0/24
192.168.5.5              0        100 10 I
10.10.3.0/24
192.168.5.5              0        100 10 I
10.10.4.0/24
192.168.5.5              0        100 10 I
10.30.1.0/24
192.168.5.5              0        100 30 I
10.30.2.0/24
192.168.5.5              0        100 30 I
10.30.3.0/24
192.168.5.5              0        100 30 I
10.30.4.0/24
192.168.5.5              0        100 30 I
```

Chablis already had reachability to 192.168.5.5 to form the peer session. Therefore, the received routes are no longer `hidden` and now appear in the local routing table with the proper `Protocol Nexthop` listed:

```
user@Chablis> show route protocol bgp terse

inet.0: 21 destinations, 22 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A Destination        P Prf   Metric 1   Metric 2  Next hop        AS path
* 10.10.1.0/24       B 170        100          0 >at-0/0/0.100    10 I
* 10.10.2.0/24       B 170        100          0 >at-0/0/0.100    10 I
* 10.10.3.0/24       B 170        100          0 >at-0/0/0.100    10 I
* 10.10.4.0/24       B 170        100          0 >at-0/0/0.100    10 I
* 10.20.1.0/24       B 170        100          0 >at-0/0/0.100    I
* 10.20.2.0/24       B 170        100          0 >at-0/0/0.100    I
* 10.30.1.0/24       B 170        100          0 >at-0/0/0.100    30 I
* 10.30.2.0/24       B 170        100          0 >at-0/0/0.100    30 I
* 10.30.3.0/24       B 170        100          0 >at-0/0/0.100    30 I
* 10.30.4.0/24       B 170        100          0 >at-0/0/0.100    30 I

user@Chablis> show route 10.10.1/24 detail

inet.0: 21 destinations, 22 routes (21 active, 0 holddown, 0 hidden)
10.10.1.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Source: 192.168.5.5
```

```
Nexthop: via at-0/0/0.100, selected
Protocol Nexthop: 192.168.5.5 Indirect nexthop: 8463088 56
State: <Active Int Ext>
Local AS:    20 Peer AS:    20
Age: 1:30:07    Metric: 0       Metric2: 1
Task: BGP_20.192.168.5.5+179
Announcement bits (2): 0-KRT 4-Resolve inet.0
AS path: 10 I
Localpref: 100
Router ID: 192.168.5.5
```

Summary

In this chapter, we looked at some of the reasons BGP was created: connectivity of AS networks, policy control, reliable transport, and scalability of the Internet. We then discussed how BGP routers form peer relationships and how routes are advertised between EBGP and IBGP peers. Following that was a discussion of the BGP message types exchanged between two routers.

We then examined how a BGP router stores its routes in various RIBs. The routes in the Adjacency-RIB-In are then parsed through a route selection algorithm to find the best path to each destination. We then examined the attributes used in the algorithm.

Finally, we explored how to configure and operate BGP on a Juniper Networks router. We explained various `show` commands used to verify connectivity. We then discussed how to source routes into BGP using a routing policy and talked about CLI commands used to view routing knowledge. We ended with an examination of a Next Hop reachability problem and one potential solution.

Exam Essentials

Describe why BGP is used for interdomain routing. BGP provides the Internet with the ability to connect AS networks in a mesh environment, allows for explicit policy control, and reliably transmits routing knowledge across the Internet.

Identify the different forms of BGP peering. A BGP router can peer with a neighbor in a different AS using EBGP. It may also peer with other routers in its own AS using IBGP.

Describe the BGP message types. There are four message types used in BGP. They include the Open, Update, Notification, and Keepalive messages.

Identify the various routing information bases used by BGP. There are three main databases used to store BGP routes: the Adjacency-RIB-In, the Local-RIB, and the Adjacency-RIB-Out tables.

List the steps of the BGP route selection algorithm. Each BGP router uses only one path advertisement to a destination. That singular path is located using a specific set of tiebreaking rules. Many steps include the comparison of BGP attributes.

Describe the major BGP attributes and their functions. Some of the BGP attributes are Next Hop, Local Preference, AS Path, Origin, and Multiple Exit Discriminator.

Key Terms

Before you take the exam, be certain you are familiar with the following terms:

`Active` state	Multiple Exit Discriminator (MED)
Adjacency-RIB-In	neighbors
Adjacency-RIB-Out	Network Layer Reachability Information (NLRI)
AS Path	Next Hop
Community	Notification message
`Connect` state	Open message
`Established` state	`OpenConfirm` state
external BGP (EBGP)	`OpenSent` state
IBGP full-mesh	Origin
`Idle` state	path-vector protocol
internal BGP (IBGP)	peers
Keepalive message	Routing Information Base (RIB)
Local Preference	TCP port 179
Local-RIB	Update message

Review Questions

1. BGP uses Transmission Control Protocol (TCP) as its transport. What is the port number?
   - **A.** 176
   - **B.** 179
   - **C.** 181
   - **D.** 173

2. After the initial route exchange, does BGP send incremental or complete updates on a regular basis?
   - **A.** BGP sends incremental updates only when needed.
   - **B.** BGP sends complete updates to ensure that the routing information is accurate.
   - **C.** BGP sends both complete and incremental updates to ensure the accuracy of routing.
   - **D.** BGP does not send any updates after the initial data exchange.

3. What description best describes BGP?
   - **A.** It is a path-vector protocol.
   - **B.** It is a distance-vector protocol.
   - **C.** It is a link-state protocol.
   - **D.** It is a hybrid routing protocol.

4. What are two characteristics of a default EBGP peer session?
   - **A.** Peers belong to different AS networks.
   - **B.** Peers belong to the same AS network.
   - **C.** Peers must be directly connected.
   - **D.** Peers do not need to be directly connected.

5. What are two characteristics of an IBGP peering?
   - **A.** Peers belong to different AS networks.
   - **B.** Peers belong to the same AS network.
   - **C.** Peers must be directly connected.
   - **D.** Peers do not need to be directly connected.

6. Why must EBGP peers be directly connected, by default?
   - **A.** An IGP does not operate between AS networks.
   - **B.** The Next Hop attribute gets changed when routes are exchanged between the peers.
   - **C.** The IP TTL of BGP messages is set to 1.
   - **D.** They do not have to be directly connected.

7. What is one use of the AS Path attribute?
   - **A.** To determine if a route is active in the routing table
   - **B.** To prevent routing loops
   - **C.** To establish a BGP peer session
   - **D.** To prevent a denial-of-service attack

8. What BGP state denotes a fully operational session?
   - **A.** `OpenSent`
   - **B.** `OpenConfirm`
   - **C.** `Established`
   - **D.** `Active`

9. How do the advertisement rules differ for IBGP and EBGP peers?
   - **A.** EBGP peers advertise only active routes, while IBGP peers advertise all routes.
   - **B.** EBGP peers advertise routes learned from both EBGP and IBGP peers, while IBGP peers do not advertise IBGP-learned routes.
   - **C.** IBGP and EBGP routers advertise all routes to all peers.
   - **D.** IBGP peers advertise routes learned from both IBGP and EBGP peers, while EBGP peers do not advertise EBGP-learned routes.

10. What is the purpose of the Notification message?
    - **A.** To notify a peer that it has completed sending all the routing updates
    - **B.** To notify peers that a route's attributes are changing
    - **C.** To open a BGP connection
    - **D.** To notify a peer that an error has been detected

11. What two things about the Local Preference attribute set it apart from the other BGP attributes?
    - **A.** A lower Local Preference is preferred over a higher one.
    - **B.** A higher Local Preference is preferred over a lower one.
    - **C.** Local Preference attempts to affect inbound traffic flows.
    - **D.** Local Preference attempts to affect outbound traffic flows.

12. What command displays routes that are in the Adjacency-RIB-Out table?
    - **A.** `show route advertising-protocol bgp` ***`neighbor-address`***
    - **B.** `show route bgp advertising-protocol` ***`neighbor-address`***
    - **C.** `show route receive-protocol bgp` ***`neighbor-address`***
    - **D.** `show route bgp receive-protocol` ***`neighbor-address`***

13. What command displays routes that are in the Adjacency-RIB-In table?
    A. `show route advertising-protocol bgp` ***`neighbor-address`***
    B. `show route bgp advertising-protocol` ***`neighbor-address`***
    C. `show route receive-protocol bgp` ***`neighbor-address`***
    D. `show route bgp receive-protocol` ***`neighbor-address`***

14. What BGP attribute is checked first in the route selection algorithm?
    A. Local Preference
    B. Origin
    C. Next Hop
    D. AS Path

15. What method of resolving Next Hop reachability is considered a best practice?
    A. Using static routes
    B. Altering the attribute value
    C. Using the IGP passive option
    D. Using an export policy to advertise `Direct` routes

16. By default, what BGP attribute is evaluated only when two path advertisements are received from the same neighboring AS?
    A. Local Preference
    B. Multiple Exit Discriminator
    C. Next Hop
    D. AS Path

17. Which of the listed BGP attributes is evaluated last in the BGP Route Selection Algorithm when determining the best path advertisement for a route?
    A. AS Path
    B. Router ID
    C. IGP metric cost
    D. Preference of EBGP routes over IBGP routes

18. What is the purpose of the Keepalive message?
    A. To begin the peer relationship
    B. To request an Update message
    C. To acknowledge a Notificaton message
    D. To maintain the BGP peer session

19. What action does the local router take after sending a Notification message to a remote peer?

A. Sends a Keepalive message to the peer.

B. Sends an Update message to the peer.

C. Terminates the peer session.

D. No action is taken.

20. Why is a full-mesh peer session required for IBGP peers?

A. Because an IGP is needed to establish the peering sessions.

B. IBGP peers do not send each other IBGP-learned routing updates to prevent routing loops.

C. EBGP peers won't advertise their routes until a full mesh is established.

D. To preserve the Next Hop attribute.