BGP Protocol

Teldat-Dm 763-I

Legal Notice

Warranty

This publication is subject to change.

Teldat offers no warranty whatsoever for information contained in this manual.

Teldat is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

I Related Documents. 1

Chapter 1 BGP Protocol 2

1.1 Introduction 2
1.1.1 Protocol Mechanisms 2
1.1.2 Message Types 2

1.2 IPv6 Protocol Extensions (MP-BGP) 4
1.2.1 Parameters 4
1.2.2 Attributes 5

1.3 Route Selection Criteria 5
1.3.1 Preference (Administrative Distance) 6
1.3.2 Preference2 (tie-breaker) 6
1.3.3 Metric (MULTI_EXIT_DISC) 6
1.3.4 Metric2 (LOCAL_PREF) 6
1.3.5 AS-path 6
1.3.6 Selecting a route 6

Chapter 2 Configuring the BGP Protocol 8

2.1 Introduction 8

2.2 Configuration Commands 8
2.2.1 Router-id 8
2.2.2 Address family 9
2.2.3 Aggregate 17
2.2.4 As 18
2.2.5 As-path 18
2.2.6 As-path-set 19
2.2.7 Default-metric 19
2.2.8 Disable 19
2.2.9 Enable 19
2.2.10 Export. 20
2.2.11 Generate 20
2.2.12 Graceful-restart. 21
2.2.13 Group 22
2.2.14 Import. 32
2.2.15 Martians. 33
2.2.16 Multipath 34
2.2.17 No aggregate 34
2.2.18 No as 35
2.2.19 No as-path. 35
2.2.20 No as-path-set 35
2.2.21 No bgp 35
2.2.22 No default-metric 36
2.2.23 No export 36
2.2.24 No generate 36
2.2.25 No graceful-restart 36

2.2.26 No group . . . 37
2.2.27 No import . . . 37
2.2.28 No martians . . . 37
2.2.29 No multipath . . . 37
2.2.30 No preference . . . 37
2.2.31 Preference . . . 38
2.2.32 Vrf . . . 38
2.2.33 Exit . . . 38

Chapter 3 BGP Protocol Monitoring . . . 39

3.1 Monitoring Tools . . . 39

3.2 Monitoring Commands . . . 39
3.2.1 Aspaths . . . 39
3.2.2 Interfaces . . . 39
3.2.3 Memory . . . 40
3.2.4 Peer-info . . . 41
3.2.5 Peer-groups . . . 42
3.2.6 Reset-message-counters . . . 43
3.2.7 Restart-bgp . . . 44
3.2.8 Routes . . . 44
3.2.9 Summary . . . 46
3.2.10 Tasks . . . 47
3.2.11 Timers . . . 48
3.2.12 Vrf . . . 48
3.2.13 Exit . . . 48

Chapter 4 Examples . . . 50

4.1 Basic Example . . . 50
4.1.1 Monitoring Router 100 . . . 51
4.1.2 Monitoring Router 200 . . . 52

4.2 Example: Address assigned dynamically by the BGP peer . . . 53

4.3 Example: Behaving as a route reflector . . . 55

4.4 Example: Exchanging Routes and IPv6 Connectivity . . . 58

I Related Documents

Teldat Dm702-I – TCP-IP

Teldat Dm764-I – Route Mapping

Teldat Dm805-I – IPv6 Addressing

Chapter 1 BGP Protocol

1.1 Introduction

The Border Gateway Protocol (BGP) was established as an Internet standard in 1989 and originally defined in *RFC 1105* . It was adopted as the preferred Exterior Gateway Protocol (EGP) for interdomain routing. The current version, BGP-4, was adopted in 1995 and defined in *RFC 4271*. BGP-4 supports Classless Inter Domain Routing (CIDR) and is the most commonly used routing protocol for routing information between autonomous systems. It has proven to be easily scalable, stable and possesses the necessary mechanisms to support complex routing policies. The acronym **BGP** in this manual stands for BGP version 4.

BGP continues to evolve through the Internet standards at the IETF. As Internet routing requirements change, the BGP protocol is extended to provide the mechanisms needed to control routing information and to support the new requirements. For this reason, the RFC base has been extended by several RFCs.

1.1.1 Protocol Mechanisms

BGP uses the Transmission Control Protocol (TCP) to establish a reliable connection between two BGP ends on port 179. One TCP session is established between each pair for each BGP session. No routing information can be exchanged until the TCP session has been established. This means there has to be a working IP connectivity between each BGP end pair. For added security, MD5 signatures can be used to authenticate each TCP segment (with IPv4 only, not IPv6).

BGP is known as a vector routing protocol because it stores routing information as a combination of a destination and attributes of the path to that destination. Using *path attributes* as criteria, the protocol uses a deterministic route selection process to select the best of the multiple feasible routes. Characteristics such as delay, link use or number of hops are not considered in this process. The route selection process, discussed later on, is key in order to understand and implement BGP policies.

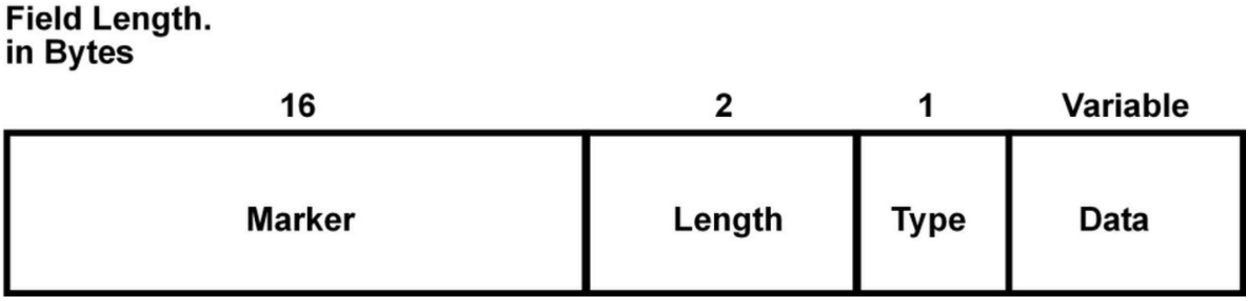
Contrary to most Interior Gateway Protocols (IGP), BGP only sends a full routing update once, at the beginning of a session, after which only incremental changes are sent. Only the routing information relative to these updates is recalculated in BGP; there is no process that must update all of its routing information like SPF calculations in OSPF or IS-IS. While IGP convergence may be faster, an IGP cannot scale to support the number of routes necessary for inter-domain routing. IGPs also lack *path attributes* that BGP carries, which are essential for selecting the best route and building routing policies. BGP is the only protocol that can be used between different autonomous systems because of the inherent support for routing policies that *path attributes* provide. These policies mean the routing information can be accepted, rejected or changed before being used to make forwarding decisions. This capacity affords network carriers a high level of protection against undesirable routing information as well as enabling them to control routing information according to their own particular needs.

BGP runs in two modes: EBGP and IBGP. EBGP (Exterior BGP) is used between routers in different autonomous systems and IBGP (Interior BGP) is used between BGP routers in the same autonomous system.

1.1.2 Message Types

BGP uses five message types to negotiate parameters, exchange routing information and indicate errors. Each message is between 19 and 4096 bytes long and relies on TCP/IP for delivery, sequencing and fragmentation. This means multiple BGP messages can be sent in one TCP segment. All messages include a common 19-byte header, with certain messages also containing additional data depending on the message type. Information in the BGP messages is usually encoded using Type-Length-Value (TLV) to provide flexibility, extensibility and ease in the processing of the messages and their data.

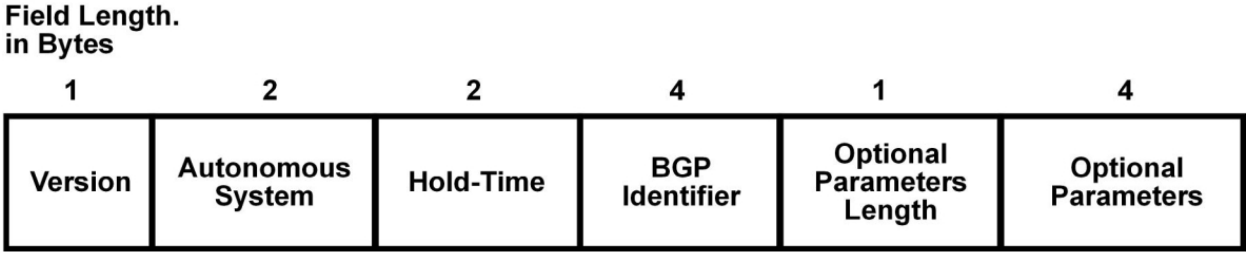
1.1.2.1 BGP message header



All BGP message types use a common header, made up of the following fields:

- *Marker* (16 bytes): Contains all ones (0xFF) and is used for synchronization when there are multiple messages in a TCP segment.
- *Length* (2 bytes): Total message length.
- *Type* (1 byte): Message type.
- *Data* (variable): Depending on the message type, data may or may not be present.

1.1.2.2 OPEN Message (Type 1 – RFC 4271)



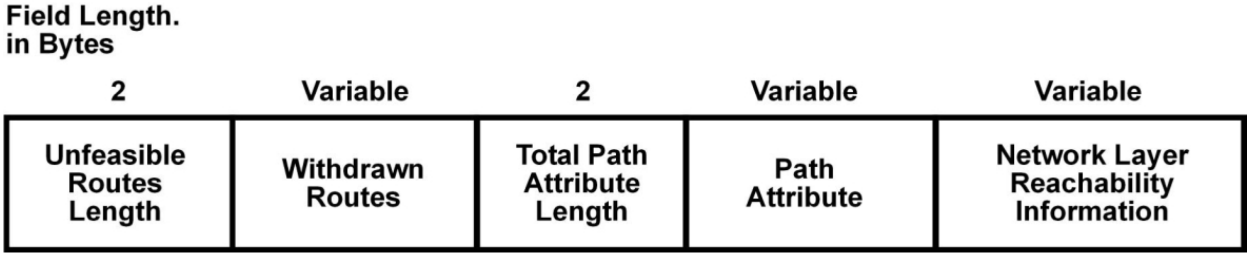
Field Length.
in Bytes

1	2	2	4	1	4
Version	Autonomous System	Hold-Time	BGP Identifier	Optional Parameters Length	Optional Parameters

The *open* message is the first message sent after TCP connection is established. This message is used to exchange configuration information and negotiate common parameters between peers. It contains the following fields:

- *Version* (1 byte): BGP version. Default is BGP-4 and cannot be changed.
- *Autonomous System* (2 bytes): AS number at the BGP end.
- *Hold Time* (2 bytes): The number of seconds that can lapse without receiving an *update* or *keepalive* message before the point-to-point connection is assumed down. Default is 180 seconds.
- *BGP Identifier* (4 bytes): Sender BGP ID, equal to the router ID.
- *Optional Parameter Length* (1 byte): Length is set to 0 if none are present.
- *Optional Parameters* (variable): Header password authentication (RFC 4271), defined in the RFC but not implemented by any manufacturer. Capabilities advertisement (RFC 2842) provides a mechanism to negotiate which optional BGP characteristics will be used.

1.1.2.3 UPDATE Message (Type 2 – RFC 4271)



Field Length.
in Bytes

2	Variable	2	Variable	Variable
Unfeasible Routes Length	Withdrawn Routes	Total Path Attribute Length	Path Attribute	Network Layer Reachability Information

Update messages in BGP are used to distribute routing information and are only sent once the session is established. An *update* message can be used to withdraw existing routes, add new routes or both.

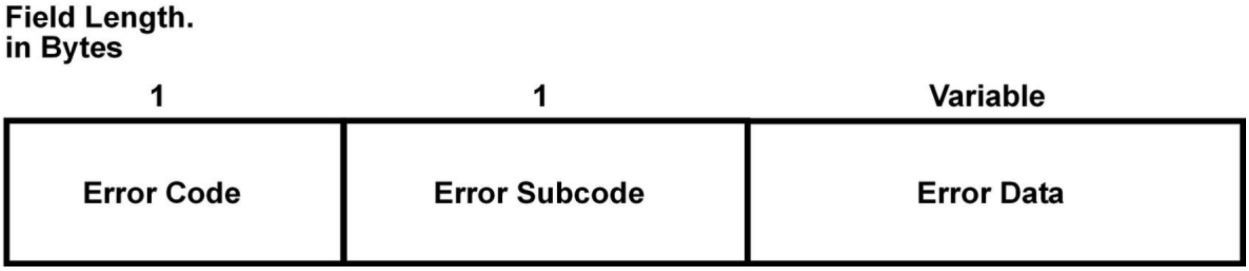
An *update* message consists of the following fields:

- *Withdrawn Routes Length* (2 bytes): Length of withdrawn routes field, a length of 0 means there are no withdrawn routes.
- *Withdrawn Routes* (variable): Routes to be withdrawn.
- *Total Path Attribute Length* (2 bytes): Length of *path attributes*. A length of 0 means there is no information.
- *Path Attributes* (variable): *Path attributes*.
- *NLRI* (Network Layer Reachability Information) (variable): IP routing prefixes.

1.1.2.4 KEEPALIVE Message (Type 3 – RFC 4271)

Keepalive messages are periodically sent to indicate the router is operating normally and to keep the BGP session up. This message only contains the header and no data.

1.1.2.5 NOTIFICATION Message (Type 4 – RFC 4271)



BGP sends a *notification* message to inform about an error. The BGP session closes once said message is sent. The error cause is sent to the other end for debugging. Error codes are defined in *RFC 4271* and tell you exactly what's wrong. A *notification* message contains the following fields:

- *Error Code* (1 byte): Type of error.
- *Error Subcode* (1 byte): Provides more details on the error.
- *Data* (variable): Optional error data.

1.2 IPv6 Protocol Extensions (MP-BGP)

From a conceptual point of view, two aspects in BGP are conditioned by the address family: connectivity (*RFC 2545*) and route communication. (*RFC 4760*).

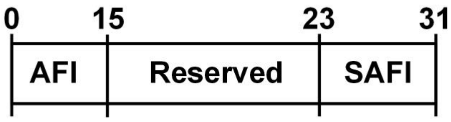
Connectivity is tied to TCP connections over which BGP messages are exchanged. With a growing shortage of Internet Protocol (IP) addresses, it became increasingly necessary to design a new version of the protocol, one that would provide a definitive solution to the limited number of available addresses (for more information, please see *Teldat Dm805-I IPv6 Addressing*). The most recent version of Internet Protocol, IPv6, plays a major role in present and future communications, which is why fundamental Internet services like BGP should support the new family. The BGP protocol itself is independent of the transport layer used. However, since extensions have made it possible to carry routes belonging to different address families, some configuration mechanisms linked to the protocol being used do exist. Later we will take a look at one of the fields directly influenced by this concept, the *Next Hop* field, which must be explicitly configured when this condition is met.

New mechanisms that solve the protocol's limitations must be incorporated now that it is possible to exchange routing information belonging to different families. There are basically three elements that are IPv4 specific, namely, the *Next Hop*, *Aggregator* and *NLRI attributes*. Hence, only two things must be added to the protocol to allow BGP to carry routes belonging to different address families: the ability to associate a particular network layer protocol with the next hop information and the ability to associate a particular network layer protocol with the *NLRI attribute*.

Another important concept introduced with IPv6 is the address scope (address scopes are described in *Teldat Dm805-I IPv6 Addressing*). There are several types of scopes but only two are identified in BGP: **globa**l and **link-local**. Globally scoped addresses do not have any specific routing limitations; they are globally routable. This is not the case with link-local addresses. A link-local address is only valid for communications within a local network segment or a point-to-point connection (routers do not forward packets with link-local addresses). Thus, the use of an IPv6 address must be appropriate for its scope in terms of both connectivity and the exchange of routes.

1.2.1 Parameters

As described in the corresponding section, an *open* message is used to establish the BGP session and to negotiate common parameters. One of the difficulties the transition to IPv6 poses involves informing neighbors in an address family of the routes to be exchanged. *RFC 4760* defines an optional parameter called *Multiprotocol Extensions capabilities* for this task. Its structure is as follows:



The meaning of each field is as follows:

- *AFI* or *Address Family Identifier* (16 bits). This field provides information on one of the possible address families (IPv4 or IPv6) supported by the neighbor.
- *SAFI* or *Subsequent Address Family Identifier* (8 bits). This field defines one of the route types (unicast or multicast) supported by the neighbor.

Now that a neighbor can exchange routes from different families in the same BGP session, more than one *Capability* can be sent during the negotiation phase. And where two neighbors exchange routes belonging to the same family, both parties must send the corresponding parameter. In the absence of this parameter, functioning is traditional (i.e.

only IPv4 routes are exchanged).

1.2.2 Attributes

There are two new attribute types that can be sent within an *update* message. Their purpose is to inform a neighbor on the availability of a route without being tied to a particular address family.

1.2.2.1 Attribute MP_REACH_NLRI (Type 14)

This is an optional *non-transitive attribute* that can be used for the following purposes:

(1) To advertise a feasible route to a peer.

(2) To allow a router to advertise the network layer address of the router that should be used as next hop to the destinations listed in the *Network Layer Reachability Information* field.

The attribute is encoded as follows:

Address Family Identifier (2 bytes)
Subsequent Address Family Identifier (1 byte)
Length of Next Hop Network Address (1 byte)
Network Address of Next Hop (var)
Reserved (1 byte)
Network Layer Reachability Information (var)

These fields have the following purpose:

- *Address Family Identifier* (2 bytes) and *Subsequent Address Family Identifier* (1 byte). These fields identify the family and address type to which the *Next Hop* and *Network Layer Reachability Information* fields belong. Typical values are 1 (IP) or 2 (IPv6) for the AFI field and 1 (unicast) or 2 (multicast) for the SAFI field.
- *Length of Next Hop Address* and *Network Address of Next Hop* (variable). This contains the length and network address of the next router on the path to the destination system. This is a variable length field, meaning it can potentially transport more than one next hop.
- *Network Layer Reachability Information* (variable). This groups all routes being advertised in an attribute.Entries for this field are <*length, value*>.

The *Next Hop* field requires special consideration. Since BGP distinguishes between two types of scopes for IPv6 addresses, the next hop propagation must verify these limitations. That is why a link-local address can only be used as the next hop in neighbors whose interfaces are connected to the same network segment (in indirectly connected peers, this is not allowed).

Another important aspect associated with this field is the sending of routes from a family other than the connectivity family. In these cases, since the value of the field is not directly deductible, you must define it explicitly (the mechanisms provided by our routers are described in the configuration section).

It is also worth noting that, although this new parameter allows IPv4 routes to be sent using the new attributes, in practice, traditional methods of communicating routes (through the *NLRI* and *Withdrawn Routes* fields in the *update* messages) are still being used.

1.2.2.2 Attribute MP_UNREACH_NLRI (Type 15)

This is an optional *non-transitive attribute* used to inform a neighbor that the previously advertised routes are no longer valid. As such, it is the opposite of the attribute described in the previous section. It is encoded as follows:

Address Family Identifier (2 bytes)
Subsequent Address Family Identifier (1 byte)
Withdrawn Routes (var)

The meaning of the first two fields is similar to the *mp_reach_nlri attribute*, but it is only applied to the *Withdrawn Routes* field. This field, in turn, is made up of <*length, value*> pairs (with each entry being a route that is no longer valid).

1.3 Route Selection Criteria

BGP works with a private routing table that includes both the routes from the router's active routing table and the routes the BGP has learned from neighbors.

There may be several routes in the BGP routing table with the same destination. Only those with the highest priority are chosen for installation in the router's active routing table. BGP employs a number of different parameters to determine the priority of each route.

Given that BGP can work with routes belonging to different families, it is important to note the families are independent from each other when it comes to route selection. This is because routes belonging to different protocols cannot be compared.

The following sections describe parameters BGP uses to select routes.

1.3.1 Preference (Administrative Distance)

Route preference corresponds to the Administrative Distance between the routing protocols on the router. This parameter has priority when selecting a route to install in the router's active routing table.

Each protocol has a default preference value. These values are summarized in the following table:

Preference	Routing protocol
0	Directly connected routes.
10	OSPF protocol (Open Shortest Path First).
60	Static routes.
100	RIP (Routing Information Protocol).
150	External OSPF routes.
170	BGP (Border Gateway Protocol).

You can set the preference value in BGP with commands such as **preference**, **import**, **peer**, etc.

1.3.2 Preference2 (tie-breaker)

The preference2 parameter, also called *tie-breaker*, is used to resolve conflicts between two routes with the same degree of preference.

Set this parameter through **peer <peer-address> preference2 <pref>**.

1.3.3 Metric (MULTI_EXIT_DISC)

The metric indicates the cost of the route and can only be compared with routes belonging to the same protocol.

The significance of the metric is defined for each protocol. In RIP, for example, it indicates the number of hops to the destination.

The metric in BGP inherits the *multi_exit_disc attribute* value.

Commands such as **default-metric**, **export** and **peer** can be used to set the metric value in BGP.

1.3.4 Metric2 (LOCAL_PREF)

In BGP this parameter inherits the value of the *local-pref attribute*. This parameter has priority if you have not assigned a value (**none** appears).

Use the **entry <n> set local-preference <local-pref>** route-map command to set this parameter.

1.3.5 AS-path

In a BGP-learned route, the AS-path indicates the autonomous systems through which the route has been learned.

1.3.6 Selecting a route

BGP uses the following criteria to select the best route, or hop, to a particular destination:

- Route with the lowest preference value (*Administrative Distance*) is chosen.
- Route with the lowest preference2 (*tie-breaker*) value is selected when there are two routes with the same preference.
- A route with the highest metric2 (*LOCAL_PREF*) value is preferred. If there is no **metric2** value (**none** is shown), the maximum value is taken.

- A route containing AS-path information is preferred over a route without AS-path information.
- When two routes have different AS-paths, the route with the shortest AS-path is preferred.
- When two routes contain different AS-path information, BGP prefers the route received from IGP. If there is no route from IGP, the route received from EGP is selected. In other words, the route with the lowest *ORIGIN* attribute value is preferred.

  The following table summarizes the *ORIGIN* attribute values defined in RFC 4271:

Value	Meaning
0	IGP - Network Layer Reachability Information is interior to the originating AS.
1	EGP - Network Layer Reachability Information learned via the EGP protocol [RFC904].
2	INCOMPLETE - Network Layer Reachability Information learned by some other means.

- When two routes contain AS-path and metric information from the same autonomous system (*AS*), the route with the lowest metric value (*MULTI_EXIT_DISC*) is preferred.
- When eiBGP multipath is disabled, a route received from an external peer (eBGP) is preferred over a route received from an internal peer (iBGP).
- Routes eligible for installation in the router's active routing table are preferred over routes that are not eligible for installation in the table.
- A route with next hop is preferred over a route with no next hop.
- If both routes have next hop and are eligible for installation in the active routing table, then they can form a multipath.
- When eiBGP multipath is enabled, a route received from an external peer (eBGP) is preferred over a route received from an internal peer (iBGP).
- The route with the lowest next hop IP address value is preferred.

The following section lists the route selection criteria on older OS versions.

1.3.6.1 Deprecated route selection criteria

The following BGP route selection criteria apply to OS versions up to and including 10.09.26, 10.09.24.20.07, 11.00.05 and 11.01.00:

- The route with the lowest preference value (*Administrative Distance*) is chosen.
- The route with the lowest preference2 (*tie-breaker*) value is selected when there are two routes with the same preference.
- A route with the highest metric2 (*LOCAL_PREF*) value is preferred. If there is no **metric2** value (**none** is shown), the maximum value is taken.
- A route containing AS-path information is preferred over a route without AS-path information.
- When two routes contain AS-path and metric information from the same autonomous system (*AS*), the route with the lowest metric value (*MULTI_EXIT_DISC*) is preferred.
- When two routes contain different AS-path information, BGP prefers the route received from IGP. If there is no route from IGP, the route received from EGP is selected.
- When two routes have different AS-paths but the same origin, the route with the shortest AS-path is preferred.
- When eiBGP multipath is disabled, a route received from an external peer (eBGP) is preferred over a route received from an internal peer (iBGP).
- Routes eligible for installation in the router active routing table are preferred over routes that are not eligible for installation in the table.
- A route with next hop is preferred over a route with no next hop.
- If both routes have next hop and are eligible for installation in the active routing table, then they can form multipath.
- When eiBGP multipath is enabled, a route received from an external peer (eBGP) is preferred over a route received from an internal peer (iBGP).
- The route with the lowest next hop IP address value is preferred.

Chapter 2 Configuring the BGP Protocol

2.1 Introduction

The incorporation of IPv6 into BGP introduces new challenges when configuring our routers. The old configuration methods are simply not expressive enough now that it's possible to work with both neighbors and routes belonging to two different families (IPv4 or IPv6).

As explained in the section on *extensions for IPv6 support*, we have two concepts that belong to different domains: connectivity and route exchange. The first of these is dealt with thanks to the **group** command. Since this command is used to define both the neighbors with whom you want to initiate a BGP session (**peer** command), and those with permission to initiate the session (**allow** command), you can now define IPv6 addresses in them. In the case of route exchange, now that routes belonging to different families can be exchanged, the **address-family** command has been added.

While this and previous commands are explained in the relevant sections, it is worth noting here the ease with which the latter command allows route management policies to be defined. The new configuration mechanism means the policies are grouped by families, which are independent of each other. For compatibility reasons, BGP can be configured in the traditional way when working with IPv4 routes (although we recommend using the new mechanisms).

The steps required to configure BGP are as follows:

- Mandatory: set the router identifier (**router-id** command from the IP section or the BGP menu itself).
- Mandatory: enable BGP (**enable** command).
- Mandatory: set the router autonomous system number (**as** command).
- Mandatory: define the BGP connections (**group** command).
- Optional: defines the route management policies per family (commands are in the corresponding address-family section). IPv4 policies are accessible from the main BGP menu for compatibility reasons.

  The commands available when defining the route management policies are: **export**, **import**, **aggregate**, **generate** and **martians**. If you want to enable IPv6 route management, you must also enable **unicast-routing** from IPv6 configuration menu (for more information, please see *Teldat Dm805-I IPv6 Addressing*).
- Optional: enable multipath route installation (**multipath** command).

A full description of each command is given in the following sections.

2.2 Configuration Commands

2.2.1 Router-id

The **router-id** command assigns the *BGP Identifier* value. This is an unsigned 32-bit integer placed in *open* messages and exchanged between two neighbors at the beginning of a BGP session. This value must be unique in the network and has the same syntax as an IPv4 address, regardless of the network protocol you are going to use.

The **router-id** command can be configured in different places:

- With the addition of extensions allowing BGP to support multiple network protocols, it's now possible to operate without relying on IPv4. This means the **router-id** command is available from the BGP menu.
- If this value is not configured in the previous menu, the value configured in the IPv4 menu is used. If this command was not used to configure the router identifier (or if an IP address not assigned to an interface is configured), the address configured through the IP **internal-ip-address** command is used. These commands are described in *Teldat Dm702-I TCP-IP Configuration*.

Syntax:

The **router-id** command syntax remains the same regardless of where you are configuring it (it has the same syntax as an IPv4 address).

```
BGP config>router-id <ip-router>
```

ip-router	IPv4 address identifying the router in BGP connections.

Example:

```
IP config>router-id 172.24.78.116
IP config>
```

2.2.2 Address family

Allows you to define route management policies per address family. This is suitable when configuring IPv4 family policies and the only possible method for an IPv6 family.

Syntax:

```
BGP config>address-family ipv4 | ipv6
```

ipv4	Route management policies for an IPv4 address family.
ipv6	Route management policies for an IPv6 address family.

Example:

```
BGP config>address-family ipv4
-- BGP IPv4 address family configuration –
BGP IPv4 config>
```

The following commands are available in this configuration menu:

Command	**Function**
aggregate	Defines an aggregation policy.
enable-family	Enables route management in current address family.
export	Defines a route export policy.
generate	Defines a route generation policy.
import	Defines a route import policy.
martians	Defines a route martians policy.
no aggregate	Eliminates an aggregation policy configuration.
no enable-family	Disables route management in current address family.
no export	Eliminates an export policy configuration.
no generate	Eliminates a generation policy configuration.
no import	Eliminates an import policy configuration.
no martians	Eliminates martians policy configuration.
no redistribute	Disables route redistribution.
redistribute	Allows you to modify attributes while IPv4 family routes are being installed.
exit	Finalizes configuration for current family.

Since the commands differ based on the syntax determined by their address family, the following sections explain the above commands per family.

2.2.2.1 Address-family ipv4

This configuration menu is simply a new, more suitable mechanism for defining route management policies for an IPv4 family. The commands duplicated in this menu are those associated with the family route management (**aggregate**, **export**, **generate**, **import**, **martians**, **redistribute** and their corresponding **no**). Since the commands are the same, their formal definition can be found in the relevant sections of this manual. The following summary table shows where to find them.

Command	**Function**
aggregate	Section 2.3.
export	Section 2.10.
generate	Section 2.11.
import	Section 2.13.
martians	Section 2.14.
no aggregate	Section 2.16.

no export	Section 2.22.
no generate	Section 2.23.
no import	Section 2.25.
no martians	Section 2.26.

2.2.2.1.1 Enable-family

Allows IPv4 family routes to be installed in the BGP table and negotiated with other neighbors. This command is default.

Syntax:

```
BGP IPv4 config>enable-family
```

Example:

```
BGP IPv4 config>enable-family
BGP IPv4 config>
```

2.2.2.1.2 No enable-family

Prevents IPv4 family routes from being installed in the BGP table and negotiated with other neighbors. Routes between neighbors are not exchanged while this command is configured, regardless of whether there are valid import or export rules.

Syntax:

```
BGP IPv4 config>no enable-family
```

Example:

```
BGP IPv4 config>no enable-family
BGP IPv4 config>
```

2.2.2.1.3 No redistribute

Disables redistribution of IPv4 family routes.

Syntax:

```
BGP IPv4 config>no redistribute
```

2.2.2.1.4 Redistribute

Allows you to modify *route attributes* during IPv4 family routes installation. Routes can be filtered according to their type and/or through route map mechanisms.

For purposes of consistency between IP and BGP tables, you can only alter attributes belonging to the latter. Modifiable attributes are shown in the following table:

	Attribute
Match	destination/mask next hop origin communities
Set	origin communities

Syntax:

```
BGP IPv4 config>redistribute {all}|{direct}|{static}|{rip}|{ospf} route-map <name>
```

Example:

We want to modify the *origin attribute* on all static routes originating from the current BGP node. These routes pass from incomplete (origin of route is unknown) to IGP (routes originating from current autonomous system). To do this, first define a **route-map** whose **set.origin** has no **match** clauses (no route dropped) and configure a single **set** clause (responsible for modifying the *origin attribute*).

Finally use the **redistribute** command to associate the route-map you created with IPv4 family static routes.

```
Config$feature route-map
Route map config$route-map set.origin
Route map set.origin$entry 1 set origin igp
Route map set.origin$exit
Route map config$exit
Config$protocol bgp
BGP config$address-family ipv4
BGP IPv4 config$redistribute static route-map set.origin
```

2.2.2.2 Address-family ipv6

Allows you to define route management policies for IPv6 family addresses. Unlike the **address-family ipv4** command, commands defined in this menu are not duplicated anywhere else. The syntax of these commands is similar to IPv4 family commands, simply replacing the addresses with those used by the IPv6 family.

2.2.2.2.1 Aggregate

Defines route aggregation policies applied to IPv6 family addresses.

Note

Its function and purpose are identical to the IPv4 family version, but this command applies to IPv6 family routes (please see the section on the IPv4 command for more information).

Syntax:

```
BGP IPv6 config>aggregate <ip6-agg-network> | default
                     [brief]
                     [as <as> | aspath <aspath> [origin [egp] [igp] [incomplete]]]
                     {{<ip6-sbn-network> [exact | refines]} |
                       all |
                       default |
                      {host <ip6-hst-address>}}
                     [{preference <pref>} | restrict]
```

`ip6-agg-network`	IPv6 aggregate network.
default	Specifies default route (route to the ::/0 network).
brief	Makes the AS path be truncated to the longest common AS path. Default is to build an AS path consisting of SETS and SEQUENCES of all contributing AS paths.
as	Restricts selection of routes to those learned from the specified Autonomous System (AS).
aspath	Restricts selection of routes to those matching the specified AS path. See **aspath** and **aspath-set** commands.
origin	Selects routes according to the origin from which they were learned.
egp	Selects routes learned through an exterior routing protocol, which does not support AS paths (for example, EGP).
igp	Selects routes learned through an interior routing protocol.
incomplete	Selects routes with incomplete AS path information.
ip6-sbn-network	IPv6 destination network address for routes to be selected.
exact	The destination mask must match the supplied mask exactly.
refines	The destination mask must be longer than the supplied mask.
all	Selects all routes.
default	Selects default routes.
host	Selects routes to the specified host through IPv6 address *ip6-hstaddress*.
preference	Specifies preference **pref** to assign to the aggregate route.
restrict	Indicates these routes are not to be considered as contributors of the specified aggregate.

Example:

```
BGP IPv6 config>aggregate 2001:db8::/32 2001:db8:1::/48 refines
BGP IPv6 config>
```

2.2.2.2.2 Enable-family

Allows IPv6 family routes to be installed in the BGP table and negotiated with other neighbors. This command is configured by default.

Syntax:

```
BGP IPv6 config>enable-family
```

Example:

```
BGP IPv6 config>enable-family
BGP IPv6 config>
```

2.2.2.2.3 Export

Defines export route policies applied to an IPv6 address family.

> **Note**
>
> Its function and purpose are identical to the IPv4 family version, but this command applies to IPv6 family routes (please see the section on the IPv4 command for more information).

Syntax:

```
BGP IPv6 config>export as <as> prot aggregate | all | direct | static | rip | ospf |
                        {bgp [aspath <aspath> [origin [egp] [igp] [incomplete]]}
                       {{<ip6-dst-network> [exact | refines]} |
                         all |
                         default |
                        {host <ip6-hst_address>}} [{metric <metric>} | restrict]
```

as	Autonomous System routes are exported to.
prot	Defines type of route to be exported. Exporting depends on the protocol (**bgp**, **rip** or **ospf**), type (**direct**, **static** or **aggregated**) or all (**all**) family routes, regardless of the protocol they belong to.
aspath	Restricts selection of routes to those matching the specified AS path. See the **aspath** and **aspath-set** commands.
origin	Selects routes according to the origin they were learned from.
egp	Selects routes learned through an exterior routing protocol that does not support AS paths (for example, EGP).
igp	Selects routes learned through an interior routing protocol.
incomplete	Selects routes with incomplete AS path information.
default	Selects the default routes.
host	Selects routes to a specified host through IPv6 address *ip6-hstaddress*. This command is the same as defining a destination network with a mask equal to that of a host (/*128*).
metric	Specifies the metric to be advertised on routes exported through this command, i.e., the *BGP multi_exit_disc attribute* .
restrict	Routes specified through this command are not exported.

Example:

```
BGP IPv6 config>export as 200 2001:db8::/48
BGP IPv6 config>
```

2.2.2.2.4 Generate

Defines route generation policies applied to an IPv6 address family.

Note

Its function and purpose are identical to the IPv4 family version, but this command applies to IPv6 family routes (please see the section on the IPv4 command for more information).

Syntax:

```
BGP IPv6 config>generate {<ip6-gen-network> | default}
                    [no-install]
                    [as <as> | as-path <aspath> [origin [egp] [igp] [incomplete]]]
                    {<ip6-sbn-network> [exact | refines] |
                     all |
                     default |
                     host <ip6-hst-address>}
                    [preference <pref> | restrict]
```

ip6-gen-network	Generated network address.
default	Specifies default route (route to the ::/0 network).
no-install	Does not add a generated route to the router active routing table.
as	Restricts selection of routes to those learned from specified Autonomous System (AS).
as-path	Restricts selection of routes to those matching specified AS path. See the **aspath** and **aspath-set** commands.
origin	Selects routes based on the origin from which they were learned.
egp	Selects routes learned through an exterior routing protocol that does not support AS paths (for example, EGP).
igp	Selects routes learned through an interior routing protocol.
incomplete	Selects routes with incomplete AS path information.
ip6-sbn-network	Destination network address for routes to select.
exact	The destination mask must match supplied mask exactly.
refines	The destination mask must be longer than the supplied mask.
all	Selects all routes.
default	Selects default routes.
host	Selects routes to the specified host through IPv6 address *ip6-hstaddress*.
preference	Specifies preference (**pref)** to assign to the generated route.
restrict	Routes specified through this command are not selected to generate this route.

Example:

```
BGP IPv6 config>generate default 2001:db8::/48
BGP IPv6 config>
```

2.2.2.2.5 Import

Defines IPv6 family routes to be added to an IPv6 routing table.

Note

Its function and purpose are identical to the IPv4 family version, but this command applies to IPv6 family routes (please see the section on the IPv4 command for more information).

Syntax:

```
BGP IPv6 config>import {as <num>} | {as-path <aspath> [origin [egp] [igp] [incomplete]]}
                  {<ip6-dst-network> [exact | refines]} |
                    all |
                    default |
                  {host <ip6-hst-address>} [preference <pref> | restrict]}
```

as	Restricts selection of routes to those learned from specified Autonomous System (AS).
as-path	Restricts selection of routes to those matching the specified AS path. See the **as-**

	path and **aspath-set** commands.
origin	Selects routes according to the origin they were learned from.
egp	Selects routes learned through an exterior routing protocol that does not support AS paths (fro example, EGP).
igp	Selects routes learned through an interior routing protocol.
incomplete	Selects routes with incomplete AS path information.
ip6-dst-network	Destination network IPv6 address for routes to be imported.
exact	The destination mask must match the supplied mask exactly.
refines	The destination mask must be longer be than the supplied mask.
all	Selects all routes.
default	Selects default routes.
host	Selects routes to the specified host through IPv6 address *ip6-hstaddress*.
preference	Specifies preference **(pref)** to assign to the imported route.
restrict	Routes specified through this command are not imported.

Example:

In this example, the following routes (belonging to an IPv6 address family) are imported from the following autonomous systems:

- *AS 100*: all routes, given there is no **import as-path** command and no **import as 100** command.
- *AS 200*: only routes to network 2001:db8::/32, as specified by command **import as 200**.
- *AS 300*: no route, specified by command **import as 300**.

```
BGP IPv6 config>import as 200 2001:db8::/32
BGP IPv6 config>import as 300 all restrict
BGP IPv6 config>
```

Example:

In this example, the following routes (belonging to an IPv6 address family) are imported from the following autonomous systems:

- *AS 100*: no route, since the **import as-path** command does not match and there is no **import as 100** command.
- *AS 200*: only routes to network 2001:db8::/32, since the import as-path command does not match and **import as 200** only matches routes to that network.
- *AS 300*: all direct routes (those that do not pass through other autonomous systems) as specified through the **import as-path pathA** command and given that there is no **import as 300** command.

```
BGP IPv6 config>as-path pathA 300
BGP IPv6 config>import as-path pathA all
BGP IPv6 config>import as 200 2001:db8::/32
BGP IPv6 config>
```

2.2.2.2.6 Martians

Defines IPv6 addresses to discard, meaning the routing information to these addresses is ignored.

Note

Its function and purpose are identical to the IPv4 family version, but this command applies to IPv6 family routes (please see the section on the IPv4 command for more information).

Syntax:

```
BGP IPv6 config>martians {{<ip6-dst-network> [exact | refines]} |
                      default |
                     {host <ip6-hst-address>}} [allow]
```

ip6-dst-address	Destination network IPv6 address to ignore.
exact	The destination mask must match the supplied mask exactly.
refines	The destination mask must be longer than the supplied mask.
default	Selects network::/0.
host	Selects the specified host address through *ip6-hst-address*.

allow	Explicitly allows the addresses indicated.

Example:

```
BGP IPv6 config>martians 2001::/16
BGP IPv6 config>
```

2.2.2.2.7 No aggregate

Removes **aggregate** command configuration for an IPv6 address family.

Syntax:

The syntax is identical to that of the **aggregate** command. Parameters must match those of a preconfigured **aggregate** command.

Example:

```
BGP IPv6 config>no aggregate 2001:db8::/32 2001:db8:1::/48 refines
BGP IPv6 config>
```

2.2.2.2.8 No enable-family

Prevents IPv6 family routes from being installed in the BGP table and negotiated with other neighbors. Routes between neighbors are not exchanged while this command is configured, regardless of whether there are valid import or export rules.

Syntax:

```
BGP IPv6 config>no enable-family
```

Example:

```
BGP IPv6 config>no enable-family
BGP IPv6 config>
```

2.2.2.2.9 No export

Removes **export** command configuration for an IPv6 address family.

Syntax:

The syntax is identical to that of the **export** command. The parameters must match those of a previously configured **export** command.

Example:

```
BGP IPv6 config>no export as 200 2001:db8::/32
BGP IPv6 config>
```

2.2.2.2.10 No generate

Removes **generate** command configuration for an IPv6 address family.

Syntax:

The syntax is identical to that of the **generate** command. Parameters must match those of a preconfigured **generate** command.

Example:

```
BGP IPv6 config>no generate default 2001:db8::/32
BGP IPv6 config>
```

2.2.2.2.11 No import

Removes **import** command configuration for an IPv6 address family.

Syntax:

The syntax is identical to that of the **import** the command. Parameters must match those of a preconfigured **import** command.

Example:

```
BGP IPv6 config>no import as 200 all
BGP IPv6 config>
```

2.2.2.2.12 No martians

Removes **martians** command configuration for an IPv6 address family.

Syntax:

Th syntax is identical to that of the **martians** command. Parameters must match those of a preconfigured **martians** command.

Example:

```
BGP IPv6 config>no martians 2001:db8::/32
BGP IPv6 config>
```

2.2.2.2.13 No redistribute

Disables redistribution of IPv6 family routes.

Syntax:

```
BGP IPv6 config>no redistribute
```

2.2.2.2.14 Redistribute

Allows you to modify *IPv6 family route attributes* during route installation.

Note

Its function and purpose are identical to the IPv4 family version, but this command applies to IPv6 family routes (please see the section on the IPv4 command for more information).

Syntax:

```
BGP IPv6 config>redistribute {all}|{direct}|{static}|{rip}|{ospf} route-map <name>
```

2.2.2.2.15 Exit

Finalizes group configuration.

Syntax:

```
BGP IPv6 config>exit
```

Example:

```
BGP IPv6 config>exit
BGP config>
```

2.2.2.3 No address-family ipv4

Removes route management policies associated to IPv4.

Syntax:

```
BGP IPv4 config>no address-family ipv4
```

Example:

```
BGP IPv4 config>no address-family ipv4
BGP IPv4 config>
```

2.2.2.4 No Address-family ipv6

Removes route management policies associated to IPv6.

Syntax:

```
BGP IPv6 config>no address-family ipv6
```

Example:

```
BGP IPv6 config>no address-family ipv6
BGP IPv6 config>
```

2.2.3 Aggregate

Defines the route aggregate policies applied to an IPv4 address family. Route aggregation is a mechanism that helps create a more general route taking a specific one as model. The router only performs aggregation that has been explicitly configured through the **aggregate** command. Routes aggregated through this command can only be used in BGP connections when advertising routes. The router does not use aggregate routes to forward packets.

> **Note**
>
> Since this command applies to routes belonging to an IPv4 address family, we recommend configuring it through its equivalent command in the *address-family ipv4* menu.

Syntax:

```
BGP config>aggregate {<agg-address> mask <agg-mask> | default}
                     [brief]
                     [as <as> | aspath <aspath> [origin [egp] [igp] [incomplete]]]
                     {<sbn-address> mask <sbn-mask> [exact | refines] |
                      all |
                      default |
                      host <hst-address>}
                     [preference <pref> | restrict]
```

agg-address	Aggregate network IP address.
agg-mask	Aggregate network IP mask.
default	Specifies default route (route to network 0.0.0.0/0).
brief	Used to specify the AS path should be truncated to the longest common AS path. Default is to build an AS path consisting of SETS and SEQUENCES of all contributing AS paths.
as	Restricts selection of routes to those learned from a specified AS (Autonomous System).
aspath	Restricts selection of routes to those matching a specified AS path. See **aspath** and **aspath-set** commands.
origin	Selects routes according to the origin they were learned from.
egp	Selects routes learned through an exterior routing protocol that does not support AS paths (e.g. EGP).
igp	Selects routes learned through an interior routing protocol.
incomplete	Selects routes whose AS path information is incomplete.
sbn-address	Destination network IP address for routes to be selected.
sbn-mask	Destination network IP mask for routes to be selected.
exact	The destination mask must match the supplied mask exactly.
refines	The destination mask be must be longer than the supplied mask.
all	Selects all routes.
default	Selects default routes.
host	Selects routes to a specified host through ip address *hstaddress*.
preference	Specifies preference (**pref)** to assign to the aggregate route.
restrict	Indicates these routes are not to be considered as contributors of the specified aggregate.

Example:

```
BGP config>aggregate 10.0.0.0 mask 255.0.0.0 10.0.0.0 mask 255.0.0.0 refines
BGP config>
```

2.2.4 As

Defines the number of the autonomous system (AS) the router belongs to. You can indicate the maximum number of times the AS can appear in an AS path.

This latter option requires special attention. Normal BGP operation (RFC4271) specifies that routes forming loops should be dropped. A loop is formed when the same AS appears more than once in an AS path. You may have a scenario, however, where you don't want to drop a route. Fortunately, using this option, the administrator can modify default behavior according to the autonomous system requirements.

Note

Administrators are responsible for dealing with routing loops, both in terms of imported and exported routes.

Syntax:

```
BGP config>as <as> [loops <max-loops>]
```

as	Router AS number. Range is from 1 to 65535.
max-loops	Maximum number of times this AS can appear in an AS path. Valid range is from 1 to 10. Default is **1**.

Example:

```
BGP config>as 100
BGP config>
```

2.2.5 As-path

Defines an AS-path. When an AS-path is defined, it is assigned a name, which is used to refer to the path in other commands (**aspath-set**, **aggregate** and **import**).

Syntax:

```
BGP config>as-path <name> [<as> | <ref> | any] [min <min-rep> [max <max-rep>]]
```

name	Name of AS-path.
as	AS number. Valid range is from 1 to 65535.
ref	Name of another defined AS-path or AS-path-set.
any	Matches any AS number.
min-rep	Minimum number of times this element can be repeated consecutively in an AS-path. Valid range is from 0 to 100.
max-rep	Maximum number of times this element can be repeated consecutively in an AS-path. If this is not specified (only **min-rep** is specified), the maximum number of repetitions is infinite. Valid range is from 1 to 100. **max-rep** value must be greater or equal to **min-rep**. If the **min-rep** is set to 0, only a **max-rep** equal to 1 is admitted.

Example:

The following example specifies that myaspath AS-path matches the path formed by the autonomous systems 200-500 or 200-400-500.

```
BGP config>as-path "myaspath" 200
BGP config>as-path "myaspath" 400 min 0 max 1
BGP config>as-path "myaspath" 500
BGP config>
```

Note

The order in which the **asap** commands are introduced is important because they determine the order of the autonomous systems in the autonomous systems' path.

2.2.6 As-path-set

Defines a set of possible AS-paths. Each defined AS-path-set is assigned a name, which references the path in other commands (**aspath-set**, **aggregate** and **import**).

Syntax:

```
BGP config>as-path-set <name> <ref>
```

name	Name of AS-path-set.
ref	Name of another defined AS-path or AS-path-set.

Example:

The following example specifies that myaspathset AS-path-set matches the path formed by the autonomous systems 200-500 or 200-400-500.

```
BGP config>as-path "firstaspath" 200
BGP config>as-path "firstaspath" 500
BGP config>as-path "secondaspath" 200
BGP config>as-path "secondaspath" 400
BGP config>as-path "secondaspath" 500
BGP config>as-path-set "myaspathset" "firstaspath"
BGP config>as-path-set "myaspathset" "secondaspath"
BGP config>
```

2.2.7 Default-metric

Helps define the metric (*multi_exit_disc attribute* value) to be used when advertising routes via BGP. By default, metrics are not propagated. Metrics configured in connections or export policies override the value configured through **default-metric**.

Syntax:

```
BGP config>default-metric <metric>
```

Metric	Metric value when advertising routes via BGP. The valid range goes from 0 to 65535.

Example:

```
BGP config>default-metric 10
BGP config>
```

2.2.8 Disable

Disables BGP so all associated configuration is ignored.

Syntax:

```
BGP config>disable
```

Example:

```
BGP config>disable
BGP config>
```

2.2.9 Enable

Enables BGP. If this command is not specified, BGP configuration is ineffective.

Syntax:

```
BGP config>enable
```

Example:

```
BGP config>enable
BGP config>
```

2.2.10 Export

Defines route export policies applied to an IPv4 address family. Route exportation is a mechanism allowing you to configure rules to determine which routes will be exported to a given AS. Filters can be applied at protocol and routing layers. You can apply additional conditions to the latter to further expand available possibilities.

Note

Since this command applies to routes belonging to an IPv4 address family, we recommend configuring it through its equivalent command in the *address-family ipv4* menu.

Syntax:

```
BGP config>export as <as> prot aggregate | all | direct | static | rip | ospf |
                  {bgp [aspath <aspath> [origin [egp] [igp] [incomplete]]}
                  {{<dst-address> mask <dst-mask> [exact | refines]} |
                    all |
                    default |
                   {host <hst_address>}} [{metric <metric>} | restrict]
```

as	Autonomous System these routes are exported to.
prot	Defines the type of routes to be exported. Routes are exported according to the protocol (**bgp**, **rip** or **ospf**), type (**direct**, **static** or **aggregated**) or all (**all**) family routes regardless of the protocol they belong to.
aspath	Restricts selection of routes to those matching the specified AS path. See **as-path** and **aspath-set** commands.
origin	Selects routes according to the origin they were learned from.
egp	Selects routes learned through an exterior routing protocol that does not support AS paths (e.g. EGP).
igp	Selects routes learned through an interior routing protocol.
incomplete	Selects routes with incomplete AS path information.
dst-address	Destination network IP address for routes to be exported.
dst-mask	Destination network IP mask for routes to be exported.
exact	The destination mask must match the supplied mask exactly.
refines	The destination mask must be longer than the supplied mask.
all	Selects all routes from the selected protocol.
default	Selects default routes.
host	Selects routes to the specified host through ip address *hst-address.* This command is the same as defining a destination network with a mask equal to that of a host (255.255.255.255).
metric	Specifies the metric to be advertised in routes exported through this command, i.e., the *BGP multi_exit_discattribute* .
restrict	Routes specified through this command are not exported.

Example:

```
BGP config>export as 200 192.168.0.0 mask 255.255.0.0
BGP config>
```

2.2.11 Generate

Defines route aggregation policies applied to an IPv4 address family. Route generation allows you to automatically create a new route based on the presence of another. The route created inherits the next hops and AS path from the contributing route with the lowest (most favorable) preference.

Note

Since this command applies to routes belonging to an IPv4 address family, we recommend configuring it through its equivalent command in the *address-family ipv4* menu.

Syntax:

```
BGP config>generate {<gen-address> mask <gen-mask> | default}
                    [no-install]
                    [as <as> | as-path <aspath> [origin [egp] [igp] [incomplete]]]
                    {<sbn-address> mask <sbn-mask> [exact | refines] |
                     all |
                     default |
                     host <hst-address>}
                    [preference <pref> | restrict]
```

gen-address	Generated network IP address.
gen-mask	Generated network IP mask.
default	Specifies default route (route to network 0.0.0.0/0).
no-install	Do not install the generated route in the router active routing table.
as	Restricts selection of routes to those learned from a specified AS.
as-path	Restricts selection of routes to those matching a specified AS path. See the **as-path** and **aspath-set** commands.
origin	Selects routes according to the origin they are learned from.
egp	Selects routes learned by an exterior routing protocol, which does not support AS paths (e.g. EGP).
igp	Selects routes learned by an interior routing protocol.
incomplete	Selects routes with incomplete AS path information.
sbn-address	Destination network IP address for the routes to be selected.
sbn-mask	Destination network IP mask for the routes to be selected.
exact	The destination mask must match the supplied mask exactly.
refines	The destination mask must be longer than the supplied mask.
all	Selects all routes.
default	Selects default routes.
host	Selects routes to a specified host through IP address *hst-address.*
preference	Specifies preference (**pref)** to assign to the generated route.
restrict	Indicates these routes are not to be considered when generating a specified route.

Example:

```
BGP config>generate default 10.0.0.0 mask 255.0.0.0
BGP config>
```

2.2.12 Graceful-restart

This command allows you to configure the *Graceful Restart* feature in BGP. This feature is defined in RFC 4724 “Graceful Restart Mechanism for BGP”.

Syntax:

```
BGP config>graceful-restart {enable |
                             disable |
                             forwarding-state {actual | forced} |
                             restart-time <rstim> |
                             update-delay <udelay>}
```

enable	Enables *Graceful Restart*.
disable	Disables *Graceful Restart*.
forwarding-state	Defines the policy to set the **Forwarding State** (F) bit when sending a *Graceful Restart Capability* message. The **actual** option indicates the bit is set depending on whether the state of the routing table is preserved or not. The **forced** option indicates the bit is always set to 1, regardless of whether the state of the routing table has been preserved or not.
restart-time	Defines **rstim** time in seconds. Estimates the time it takes a BGP session to reestablish following a restart. This value is sent in a *Graceful Restart Capability* mes-

	sage at the beginning of a BGP session.
update-delay	Defines the maximum **udelay** time, in seconds, when the route selection process is delayed due to a *Graceful Restart* process. This time is used to ensure all sessions are reestablished following a restart.

Example:

```
BGP config>graceful-restart enable
BGP config>
```

2.2.13 Group

Accesses the configuration of a group of BGP peers in the same autonomous system.

Syntax:

```
BGP config>group type [external | internal] peer-as <as>
```

external	Group of eBGP peers in a different autonomous system.
internal	Group of iBGP peers in the same autonomous system.
as	Autonomous system used to define peers.

Example:

```
BGP config>group type external peer-as 200

-- BGP group configuration --
Group config>
```

A group's configuration menu contains the following commands:

Command	**Function**
allow	Defines neighbors allowed to initiate a BGP session.
client-to-client reflection	Reflects routes originating from clients to other router-reflector clients.
cluster-id	Cluster identifier used in router-reflector.
no allow	Removes connectivity rules defined for anonymous neighbors.
no client-to-client reflection	Does not reflect routes originating from clients to other router-reflector clients.
no cluster-id	Removes the configured cluster-id.
no option	Removes the configuration of a parameter by default.
no peer	Removes the configuration of a peer.
option	Configures parameters for the peers by default.
peer	Configures a peer.
exit	Finalizes group configuration.

The following sections describe and explain these commands.

2.2.13.1 Allow

Defines which routers in the autonomous system are allowed to initiate a BGP connection with our router. The router behaves passively (i.e., it does not initiate any BGP connections with remote routers, it only accepts incoming connections).

The **allow** rules, which are grouped within the same menu because they are connectivity terms, are applied per family.

By default, when a particular family does not have any rules, its anonymous neighbors are not allowed to initiate a connection.

Syntax:

```
Group config>allow {<peers-address> mask <peers-mask>} | <ip6-peers-address> |
                   {all [family ipv4 | ipv6]} |
                   {host <peer-address> | <ipv6-peer-address>}
```

peers-address	Network IP address allowed to initiate a BGP connection.
peers-mask	Network IP mask allowed to initiate a BGP connection.

ip6-peers-address	Network IPv6 address allowed to initiate a BGP connection.
all	All routers in the autonomous system can initiate a BGP connection.
family	Allows the **all** command to be applied to a particular family. The available families are IPv4 and IPv6.
host	Any neighbor with *peer-address* or *ipv6-peer-address* is allowed to initiate a BGP connection.

Example:

```
Group config>allow all
Group config>
```

2.2.13.2 Client-to-client reflection

Allows iBGP routes learned through a route-reflection to be forwarded to other route-reflection clients in the group. Default is **enabled**.

Syntax:

```
Group config>client-to-client reflection
```

2.2.13.3 Cluster-id

Configures the **cluster-id** sent in the reflection routes when a router is configured as a route-reflection. It allows you to specify an IP address or an integer.

Syntax:

```
Group config>cluster-id [ip | int]
```

2.2.13.4 No allow

Disables the **allow** command configuration. Configuration of all the **allow** commands in a group will be removed if no parameter is specified. If parameters are specified, they must match a pre-entered **allow** command.

Syntax:

```
Group config>no allow {<peers-address> mask <peers-mask>} | <ip6-peers-address> |
                      {all [family ipv4 | ipv6]} |
                      {host <peer-address> | <ipv6-peer-address>}
```

peers-address	Network IP address allowed to initiate a BGP connection.
peers-mask	Network IP mask allowed to initiate a BGP connection.
ip6-peers-address	Network IPv6 address allowed to initiate a BGP connection.
all	All routers in the autonomous system can initiate a BGP connection.
family	Allows the **all** command to be applied to a particular family. The available families are IPv4 and IPv6.
host	Any neighbor with *peer-address* or *ipv6-peer-address* is allowed to initiate a BGP connection.

Example:

```
Group config>no allow
Group config>
```

2.2.13.5 No client-to-client reflection

Prevents iBGP routes learned through a route-reflection client from being forwarded to other route-reflection clients in the group. This should only be configured when route-reflection clients are connected to one another.

Syntax:

```
Group config>no client-to-client reflection
```

2.2.13.6 No cluster-id

Removes the **cluster-id** configuration.

Syntax:

```
Group config>no cluster-id
```

2.2.13.7 No option

Disables the **option** command configuration. When no parameter is specified, all group options are removed. When a parameter is specified, only the settings of the specified option are removed.

Syntax:

```
Group config>no option [address-family ipv4 unicast |
                        address-family ipv6 unicast |
                        anal-retentive |
                        gateway |
                        graceful-restart {enable | disable} |
                        graceful-restart forwarding-state {actual | forced} |
                        graceful-restart restart-time |
                        hold-time |
                        ignore-first-as-hop |
                        in-route-map |
                        keep |
                        keepalives-always |
                        local-addr |
                        local-as |
                        local-interface |
                        maximum-prefix |
                        metric-out |
                        next-hop-self |
                        no-aggregator-id |
                        no-auth-check |
                        no-shared-interface |
                        no-v4-as-loop |
                        out-delay |
                        out-route-map |
                        passive |
                        preference |
                        preference2 |
                        recv-buffer |
                        remove-private-as |
                        route-reflector-client |
                        route-to-peer |
                        send-buffer |
                        send-community |
                        set-pref |
                        ttl |
                        v3-as-loop-okay] |
                        bfd-session]
```

address-family…	Removes the corresponding *address-family* option.
anal-retentive	Removes the *anal-retentive* option.
gateway	Removes the *gateway* option.
graceful-restart…	Removes the corresponding *graceful-restart* option.
hold-time	Removes the *hold-time* option.
ignore-first-as-hop	Removes the *ignore-first-as-hop* option.
in-route-map	Removes the *in-route-map* option.
keep	Removes the *keep all or keep none* option.
keepalives-always	Removes the *keepalives-always* option.
local-addr	Removes the *local-addr* option.
local-as	Removes the *local-as* option.
local-interface	Removes the *local-interface* the option.
maximum-prefix	Removes the *maximum-prefix* option.
metric-out	Removes the *metric-out* option.
next-hop-self	Removes the *next-hop-self* option.

no-aggregator-id	Removes the *no-aggregator-id* option.
no-auth-check	Removes the *no-auth-check* option.
no-shared-interface	Removes the *no-shared-interface* option.
no-v4-as-loop	Removes the *no-v4-as-loop* option.
out-delay	Removes the *out-delay* option.
out-route-map	Removes the *out-route-map* option.
passive	Removes the *passive* option.
preference	Removes the *preference* option.
preference2	Removes the *preference2* option.
recv-buffer	Removes the *recv-buffer* option.
remove-private-as	Removes the *remove-private-as* option.
route-reflector-client	Removes the *route-reflector-client* option.
route-to-peer	Removes the *route-to-peer* option.
send-buffer	Removes the *send-buffer* option.
send-community	Removes the *send-community* option.
set-pref	Removes the *set-pref* option.
ttl	Removes the *ttl* option.
v3-as-loop-okay	Removes the *v3-as-loop-okay* option.
bfd-session	Removes the *bfd-session* option.

Example:

```
Group config>no option
Group config>
```

Command history:

Release	Modification
10.08.34.05.09	The **in-delay** option is obsolete as of version 10.08.34.05.09.
10.08.43	The **in-delay** option is obsolete as of version 10.08.43.
10.09.25	The **in-delay** option is obsolete as of version 10.09.25.
11.00.00.02.08	The **in-delay** option is obsolete as of version 11.00.00.02.08.
11.00.04	The **in-delay** option is obsolete as of version 11.00.04.
11.00.05	The **local-as** option is obsolete as of version 11.00.05.
11.01.00	The **in-delay** option is obsolete as of version 11.01.00.
11.01.00	The **local-as** option is obsolete.

2.2.13.8 No peer

Disables the **peer** command's configuration. By specifying only the peer name or address (*peer-name* or *peer-address* or *ip6-peer-address* parameter), you remove the whole configuration relative to said peer. Specifying an additional option will remove the configuration relative to said option from the relevant peer.

Syntax:

```
Group config>no peer {<peer-name> | <peer-address> | <ip6-peer-address>}
                     [address-family ipv4 unicast |
                      address-family ipv6 unicast |
                      anal-retentive |
                      gateway |
                      graceful-restart {enable | disable} |
                      graceful-restart forwarding-state {actual | forced} |
                      graceful-restart restart-time |
                      hold-time |
                      ignore-first-as-hop |
                      in-route-map |
                      keep |
                      keepalives-always |
                      local-addr |
                      local-as |
                      local-interface |
                      maximum-prefix |
                      metric-out |
                      next-hop-self |
                      no-aggregator-id |
                      no-auth-check |
                      no-shared-interface |
                      no-v4-as-loop |
                      out-delay |
                      out-route-map |
                      passive |
                      password |
                      peer-addr |
                      preference |
                      preference2 |
                      recv-buffer |
                      remove-private-as |
                      route-reflector-client |
                      route-to-peer |
                      send-buffer |
                      send-community |
                      set-pref |
                      ttl |
                      v3-as-loop-okay] |
                      bfd-session]
```

peer-name	Name of peer connection profile.
peer-address	IPv4 address for peer affected.
ip6-peer-address	IPv6 address for peer affected.
address-family…	Removes the corresponding *address-family* option.
anal-retentive	Removes the *anal-retentive* option.
gateway	Removes the *gateway* option.
graceful-restart…	Removes the corresponding *graceful-restart* option.
hold-time	Removes the *hold-time* option.
ignore-first-as-hop	Removes the *ignore-first-as-hop* option.
in-route-map	Removes the *in-route-map* option.
keep	Removes the *keep all* or *keep none* option.
keepalives-always	Removes the *keepalives-always* option.
local-addr	Removes the *local-addr* option.
local-as	Removes the *local-as* option.
local-interface	Removes the *local-interface* option.
maximum-prefix	Removes the *maximum-prefix* option.
metric-out	Removes the *metric-out* option.
next-hop-self	Removes the *next-hop-self* option.
no-aggregator-id	Removes the *no-aggregator-id* option.
no-auth-check	Removes the *no-auth-check* option.
no-shared-interface	Removes the *no-shared-interface* option.
no-v4-as-loop	Removes the *no-v4-as-loop* option.
out-delay	Removes the *out-delay* option.
out-route-map	Removes the *out-route-map* option.
passive	Removes the *passive* option.
password	Removes the *password* option (only applied with IPv4 peers).
peer-addr	Removes the named peer.
preference	Removes the *preference* option.
preference2	Removes the *preference2* option.
recv-buffer	Removes the *recv-buffer* option.
remove-private-as	Removes the *remove-private-as* option.

route-reflector-client	Removes the *route-reflector-client* option.
route-to-peer	Removes the *route-to-peer* option.
send-buffer	Removes the *send-buffer* option.
send-community	Removes the *send-community* option.
set-pref	Removes the *set-pref* option.
ttl	Removes the *ttl* option.
v3-as-loop-okay	Removes the *v3-as-loop-okay* option.
bfd-session	Removes the *bfd-session* option.

Example:

```
Group config>no peer 172.24.78.116
Group config>
```

Command history:

Release	Modification
10.08.34.05.09	The **in-delay** option is obsolete as of version 10.08.34.05.09.
10.08.43	The **in-delay** option is obsolete as of version 10.08.43.
10.09.25	The **in-delay** option is obsolete as of version 10.09.25.
11.00.00.02.08	The **in-delay** option is obsolete as of version 11.00.00.02.08.
11.00.04	The **in-delay** option is obsolete as of version 11.00.04.
11.00.05	The **local-as** option is obsolete as of version 11.00.05.
11.01.00	The **in-delay** option is obsolete as of version 11.01.00.
11.01.00	The **local-as** option is obsolete.
11.01.09	The syntax corresponding to the named peer is available as of version 11.01.09.

2.2.13.9 Option

Defines default options for current group BGP connections. Default options are inherited by each neighbor in the group. If certain options are defined for a neighbor, logical OR operation is applied between the corresponding group and neighbor layer option.

Note

Since there are no explicit connections for anonymous neighbors (they are managed through the **allow** command), the options are applied exactly as defined.

Syntax:

```
Group config>option [address-family {ipv4 unicast | ipv6 unicast}]
                    [anal-retentive]
                    [gateway <gw-address> | <ip6-gw-address> | <interface>]
                    [graceful-restart {enable | disable}]
                    [graceful-restart forwarding-state {actual | forced}]
                    [graceful-restart restart-time <rstim>]
                    [hold-time <hold-time>]
                    [ignore-first-as-hop]
                    [in-route-map <in-route-map>]
                    [keep {all | none}]
                    [keepalives-always]
                    [local-addr <lcl-address> | <ip6-lcl-address>]
                    [local-as <local-as>]
                    [local-interface <lcl-interface>]
                    [maximum-prefix {
                        <max-prefix> {restart | warning-only} |
                        unlimited}]
                    [metric-out <metric>]
                    [next-hop-self]
                    [no-aggregator-id]
                    [no-auth-check]
                    [no-shared-interface]
                    [no-v4-as-loop]
                    [out-delay <out-delay>]
                    [out-route-map <out-route-map>]
                    [passive]
                    [preference <pref>]
                    [preference2 <pref2>]
                    [recv-buffer <recv-buf-size>]
                    [remove-private-as]
                    [route-reflector-client]
                    [route-to-peer]
                    [send-buffer <send-buf-size>]
                    [send-community]
                    [set-pref <set-pref>]
                    [ttl <ttl>]
                    [v3-as-loop-okay]
                    [bfd-session]
```

address-family	Enables a specified family to exchange routes with a neighbor. The available families are: *ipv4 unicast*, *ipv6 unicast*. If no family has been enabled through this command, *ipv4 unicast* family routes are exchanged.
anal-retentive	Generates warnings when dubious BGP updates (duplicated routes or deleted non-existent routes) are received.
gateway	Specifies the output router or interface for routes received. Since a neighbor can belong to two different families, two different gateway commands (one for each address family) can be defined. For IPv4 family, the **gateway** and **interface** options are exclusive, so only one of them may be configured at one particular time. If this command is configured and a **route-map** exists, the **gateway** sentence overrides the latter.
graceful-restart {enable | disable}	Enables or disables *Graceful Restart* for the corresponding neighbor.
graceful-restart forwarding-state {actual | forced}	Defines the policy to set the **Forwarding State** (F) bit when sending a *Graceful Restart Capability* message. The **actual** option indicates the bit is set according to whether the state of the routing table has been preserved or not. The **forced** option indicates the bit is always set to 1, regardless of whether the state of the routing table has been preserved or not.
graceful-restart restart-time	Defines the **rstim** time, in seconds, which estimates the time needed for a BGP session to reestablish following a restart. This value is sent in the *Graceful Restart Capability* message at the beginning of a BGP session.
hold-time	Specifies the **Hold Time** value to negotiate when starting a BGP connection. This value represents the maximum time that can lapse without receiving any messages from the other end.
ignore-first-as-hop	Allows routes from servers that do not add their own AS.
in-route-map	Lets you configure a **route-map** to apply to the routes received. For further information, please see *Teldat Dm764-I Route Map* .
keep all	Routes containing duplicate AS path entries are dropped by default. This option allows you to modify behavior so these routes can be installed in the BGP table.
keep none	Learned routes are not retained with this option.
keep-alives-always	*Keepalive* messages are sent even when *update* messages have been sent.
local-addr	Defines an address to be used as the local end in a TCP connection, thus providing a mechanism for selecting a particular network address with respect to the global group of addresses. Given that a neighbor can belong to two different families, two different **lcl-address** commands (one for each address family) can be defined.
local-as	Specifies an autonomous system used to identify the router in the connection.
local-interface	Establishes a local address (static or dynamic) to use in a TCP connection belonging to the *lclinterface* interface.
maximum-prefix	Configures the maximum number of prefixes accepted by the connection. If you configure **unlimited**, all prefixes are accepted. When configuring a value, indicate what action to perform should the value be exceeded. You can choose between restarting the connection (**restart**) or reporting the event (**warning-only**).
metric-out	Configures the *metric* for routes advertised through this connection. This value discards a value configured through the **default-metric** and **export** commands.
next-hop-self	Disables the next hop calculation when sending routes, so its own address is always given as the next hop.

no-aggregator-id	Omits router-id in an *aggregation attribute*, to prevent other AS routers from creating aggregation routes with different AS paths.
no-auth-check	The router normally checks the authentication field is *all ones*. This option can be used to allow communication with systems using other types of authentication.
no-shared-interface	Allows a BGP session to be established between neighbors belonging to different network segments.
no-v4-as-loop	Prevents propagation of routes with loops in the AS path to version 4 external peers. An AS path with a loop is one where the same AS appears more than once. This option allows you to stop routes from spreading erroneously to version 3 neighbors through version 4 peers.
out-delay	Specifies the *out-delay* time where a route must be stable before being exported to BGP.
out-route-map	Lets you configure a **route-map** to apply when sending routes. For further information on **route-map** configuration, please see *TeldatDm764-I Route Map* .
passive	Specifies the router will not initiate the BGP connection. It will wait to receive a connection attempt from the peer.
preference	Specifies a **pref** to assign to learned routes. Said preference value can be discarded in favor of a value configured through the **import** command.
preference2	Specifies a **pref2** to use in learned routes when the *preference* value matches.
recv-buffer	Configures a **recv-buf-size** byte size for a reception buffer.
remove-private-as	Removes private AS numbers from the updates.
route-reflector-client	Configures all members of a group as clients for this route reflector.
route-to-peer	Creates a route to the peer. This parameter exists for compatibility reasons.
send-buffer	Configures a **send-buf-size** byte size for a transmission buffer.
send-community	Enables sending *community attributes*.
set-pref	Applies a **local_pref** preference to connections with internal peers, mapping the **pref** preference value.
ttl	Specifies a **ttl** value (time-to-live) to use in packets sent in a TCP session.
v3-as-loop-okay	Allows routes with loops to be sent to version 3 external peers.
bfd-session	Registers the peer in a BFD session.

Example:

```
Group config>option hold-time 60
Group config>
```

Command history:

Release	**Modification**
10.08.34.05.09	The **in-delay** option is obsolete as of version 10.08.34.05.09.
10.08.43	The **in-delay** option is obsolete as of version 10.08.43.
10.09.25	The **in-delay** option is obsolete as of version 10.09.25.
11.00.00.02.08	The **in-delay** option is obsolete as of version 11.00.00.02.08.
11.00.04	The **in-delay** option is obsolete as of version 11.00.04.
11.00.05	The **local-as** option is obsolete as of version 11.00.05.
11.01.00	The **in-delay** option is obsolete as of version 11.01.00.
11.01.00	The **local-as** option is obsolete as of version 11.01.00.

2.2.13.10 Peer

Defines the address of a neighbor, either IPv4 (*peer-address)* or IPv6 (*ip6-peer-address*), with whom the BGP connection is initiated.

Peers can be named using options **peer <peer-name> peer-addr <ip4-or-ip6-peer-address> [local-addr <ip4-or-ip6-local-address>]**. This syntax allows for many connections to the same peer address from several local addresses, so that multihoming can be deployed with BGP connections that have a multipath architecture.

Note

Since default options and personalized options can be defined for a neighbor, the end value is the result of executing the logical OR operation between these two.

Syntax:

```
Group config>peer {<peer-name> | <peer-address> | <ip6-peer-address>}
                  [address-family {ipv4 unicast | ipv6 unicast}]
                  [anal-retentive]
                  [gateway <gw-address> | <ip6-gw-address> <interface>]
                  [graceful-restart {enable | disable}]
                  [graceful-restart forwarding-state {actual | forced}]
                  [graceful-restart restart-time <rstim>]
                  [hold-time <hold-time>]
                  [ignore-first-as-hop]
                  [in-route-map <in-route-map>]
                  [keep {all | none}]
                  [keepalives-always]
                  [local-addr {<lcl-address> | <ip6-lcl-address>}]
                  [local-as <local-as>]
                  [local-interface <lcl-interface>]
                  [maximum-prefix {
                      <max-prefix> {restart | warning-only} |
                      unlimited}]
                  [metric-out <metric>]
                  [next-hop-self]
                  [no-aggregator-id]
                  [no-auth-check]
                  [no-shared-interface]
                  [no-v4-as-loop]
                  [out-delay <out-delay>]
                  [out-route-map <out-route-map>]
                  [passive]
                  [password <authentication-key>]
                  [peer-addr {<peer-address> | <ip6-peer-address>}]
                  [preference <pref>]
                  [preference2 <pref2>]
                  [recv-buffer <recv-buf-size>]
                  [remove-private-as]
                  [route-reflector-client]
                  [route-to-peer]
                  [send-buffer <send-buf-size>]
                  [send-community]
                  [set-pref <set-pref>]
                  [ttl <ttl>]
                  [v3-as-loop-okay]
                  [bfd-session]
```

peer-name	Name of peer connection profile. Peers with different *peer-name* can share the same *peer-addr* with different *local-addr*.
peer-address	Neighbor IPv4 address used to initiate a BGP connection.
ip6-peer-address	Neighbor IPv6 address used to initiate a BGP connection.
address-family	Enables the family specified for exchanging routes with the neighbor. The available families are: *ipv4 unicast*, *ipv6 unicast*. If no family has been enabled through this command, *ipv4 unicast* family routes are exchanged.
anal-retentive	Causes warnings to be generated when dubious BGP updates (duplicated routes or deleted non-existent routes) are received.
gateway	Specifies an output router (*gw-address* or *ip6-gw-address)* or the output interface *(interface)* for routes received from a BGP neighbor. This address or interface is used in these routes instead of the received *next_hopattribute*. You should also use this field to check connectivity. If this command is configured and a **route-map** exists, the *gateway* sentence overrides the latter.
graceful-restart {enable | disable}	Enables or disables *Graceful Restart* for this neighbor.
graceful-restart forwarding-state {actual | forced}	Defines the policy to establish the *Forwarding State* (F) bit when sending a *Graceful Restart Capability* message to a neighbor. The **actual** option indicates the bit is set according to whether or not you have preserved the state of the routing table. The *forced* option indicates the bit is always set to 1, regardless of whether or not the state of the routing table has been preserved.

graceful-restart restart-time	Defines the **restim** time, in seconds, which estimates the amount of time it will take a BGP session to reestablish following a restart. This value is sent in a *Graceful Restart Capability* message at the beginning of a BGP session.
hold-time	Specifies a *Hold Time* value to negotiate when initiating the BGP connection. This value represents the maximum time that can lapse without receiving messages from the other end.
ignore-first-as-hop	Allows routes from servers that do not add their own AS.
in-route-map	Lets you configure a **route-map** to apply to routes received. For further information, please see *Teldat Dm764-I Route Map* .
keep all	Routes containing duplicated AS path entries are dropped by default. This option allows you to modify the behavior so that said routes can be installed in the BGP table.
keep none	Learned routes are not retained with this option.
keep-alives-always	*Keepalive* messages are sent even when *update* messages have been sent.
local-addr	Defines an address, either IPv4 (*lcl-address*) or IPv6 (*ip6-lcl-address*), to be used as the local end in a TCP connection, thus providing a mechanism for selecting a particular network address with respect to a global group of addresses.
local-as	Specifies an AS used to identify the router in the connection.
local-interface	Establishes a local address (static or dynamic) to use in a TCP connection belonging to the *lclinterface* interface.
maximum-prefix	Configures the maximum number of prefixes accepted by the connection. If you configure **unlimited**, all prefixes are accepted. When configuring a value, indicate what action to perform in the event the value is exceeded. You can choose between restarting the connection (**restart**) or reporting the event (**warning-only**).
metric-out	Configures the **metric** for routes advertised through this connection. This value discards the value configured through the **default-metric** and **export** commands.
next-hop-self	Disables next hop calculation when sending routes so its own IP address is always given as the next hop.
no-aggregator-id	Omits **router-id** in the *aggregation attribute* , to prevent other AS routers from creating aggregation routes with different AS paths.
no-auth-check	The router normally checks the authentication field is *all ones*. This option can be used to allow communication with systems using other types of authentication.
no-shared-interface	Allows a BGP session between neighbors belonging to different network segments.
no-v4-as-loop	Prevents the propagation of routes with loops in the AS path to version 4 external peers. An AS path with a loop is one where the same AS appears more than once. This option allows you to stop said routes from spreading erroneously to version 3 neighbors through version 4 peers.
out-delay	Specifies **out-delay** time where a route must be stable before being exported to BGP.
out-route-map	Lets you configure a **route-map** to apply when sending routes. For more information on **route-map** configuration, please see *Teldat Dm764-I Route Map* .
passive	Specifies a router will not initiate the BGP connection. This option is only available for external groups. For internal groups, this must be configured through the **option** command. It will wait to receive a connection attempt from the peer.
password	Configures a password so the connection with the remote peer is through MD5 authentication (only applied with IPv4 peers).
peer-addr	Specifies the neighbor address used to initiate a BGP connection for named peers.
preference	Specifies a **pref** to assign to learned routes. This preference value can be discarded in favor of the value configured through the **import** command.
preference2	Specifies a **pref2** to use in learned routes when a preference value matches.
recv-buffer	Configures a *recv–buf-size* byte size for the reception buffer.
remove-private-as	Removes the private AS numbers from the updates.
route-reflector-client	Configures this peer as the route-reflector client.
route-to-peer	Creates a route to a peer. This parameter exists for compatibility reasons.
send-buffer	Configures a *send–buf-size* byte size for transmission buffer.
send-community	Enables sending of *community attributes*. This option is only available for external groups. For internal groups, this must be configured through the **option** command.

set-pref	Applies the local_pref preference in connections with internal peers, mapping with a **pref** preference value.
ttl	Specifies a **ttl** value (time-to-live) to use in packets sent in the TCP session.
v3-as-loop-okay	Allows routes with loops to be sent to version 3 external peers.
bfd-session	Registers a peer in a BFD session.

Example:

```
Group config>peer 172.24.78.116
Group config>peer 2001:db8:1::101
Group config>
```

Note

BGP will consider the changes applied dynamically over an **in-route-map** once the **restart-bgp <peer ip>** monitoring command is applied.

Command history:

Release	Modification
10.08.34.05.09	The **in-delay** option is obsolete as of version 10.08.34.05.09.
10.08.43	The **in-delay** option is obsolete as of version 10.08.43.
10.09.25	The **in-delay** option is obsolete as of version 10.09.25.
11.00.00.02.08	The **in-delay** option is obsolete as of version 11.00.00.02.08.
11.00.04	The **in-delay** option is obsolete as of version 11.00.04.
11.00.05	The **local-as** option is obsolete as of version 11.00.05.
11.01.00	The **in-delay** option is obsolete as of version 11.01.00.
11.01.00	The **local-as** option is obsolete as of version 11.01.00.
11.01.04	The **out-route-map** and **send-community** options are available in iBGP peers as of version 11.01.04.
11.01.09	The syntax concerning the named peer that allows multihoming to be enabled is available as of version 11.01.09.

2.2.13.11 Exit

Finalizes group configuration.

Syntax:

```
Group config>exit
```

Example:

```
Group config>exit
BGP config>
```

2.2.14 Import

Defines routes belonging to an IPv4 family and found in the IP routing table. The BGP routing table explicitly marks routes that are not imported using the mechanisms described in the *monitoring* section.

The **import** command behaves as a filter:

- If you do not enter an **import** command, all routes are imported.
- If you enter at least one **import as-path** command, only routes specified through the **import** commands are imported.
- If you enter at least one **import as <as>** command from the **<as>** autonomous system, only routes specified through the **import** commands are imported.

Note

Since this command is applied to routes from an IPv4 address family, we recommend configuring it through its equivalent command in the address-family ipv4 menu.

Syntax:

```
BGP config>import {as <num>} | {as-path <aspath> [origin [egp] [igp] [incomplete]]}
                  {<dst-address> mask <dst-mask> [exact | refines]} |
                    all |
                    default |
                  {host <hst-address>} [preference <pref> | restrict]}
```

as	Restricts selection of routes to those learned from the specified AS.
as-path	Restricts selection of routes to those matching the specified AS path. See the **as-path** and **aspath-set** commands.
origin	Selects routes according to the origin from which they were learned.
egp	Selects routes learned through an exterior routing protocol that does not support AS paths (EGP for example).
igp	Selects routes learned by an interior routing protocol.
incomplete	Selects routes with incomplete AS path information.
dst-address	Destination network IP address of the routes to be imported.
dst-mask	Destination network IP mask of the routes to be imported.
exact	The destination mask must match the supplied mask exactly.
refines	The destination mask must be longer than the supplied mask.
all	Selects all routes.
default	Selects default routes.
host	Selects routes to a specified host through ip address *hst-address.*
preference	Specifies a **pref** to assign to an imported route.
restrict	Routes specified through this command are not imported.

Example:

In this example, the following routes (belonging to an IPv4 address family) are imported from the following autonomous systems:

- AS 100: all routes, as there are no **import as-path** and no **import as 100** commands.
- AS 200: only routes to network 192.168.0.0/16, specified by the **import as 200** command.
- AS 300: no route, specified by the **import as 300** command.

```
BGP config>import as 200 192.168.0.0 mask 255.255.0.0
BGP config>import as 300 all restrict
BGP config>
```

Example:

In this example, the following routes (once again belonging to an IPv4 address family) are imported from the following autonomous systems:

- AS 100: none of the routes, as the **import as-path** command does not match and there is no **import as 100** command.
- AS 200: only routes to network 192.168.0.0/16, as the **import as-path** command does not match and the **import as 200** command only matches routes to that network.
- AS 300: all direct routes (not those routes passing through other autonomous systems), as the **import as-path pathA** command specifies and there is no **import as 300** command.

```
BGP config>as-path pathA 300
BGP config>import as-path pathA all
BGP config>import as 200 192.168.0.0 mask 255.255.0.0
BGP config>
```

2.2.15 Martians

Defines IPv4 addresses to discard, meaning all routing information to these addresses is ignored. A poorly configured system will occasionally send obviously invalid destination addresses. You can use this command to reject invalid addresses, known as martians, on an exceptional basis.

Note

Since this command applies to routes belonging to an IPv4 address family, we recommend configuring it through its equivalent command in the address-family ipv4 menu.

Syntax:

```
BGP config>martians {{<dst-address> mask <dst-mask> [exact | refines]} |
                      default |
                     {host <hst-address>}} [allow]
```

dst-address	Destination network IP address to ignore.
dst-mask	Destination network IP mask to ignore.
exact	The destination mask must match the supplied mask exactly.
refines	The destination mask must be longer than the supplied mask.
default	Selects network 0.0.0.0/0.
host	Selects a specified host address through *hst-address.*
allow	Explicitly allows the addresses indicated.

Example:

```
BGP config>martians 0.0.0.0 mask 255.0.0.0
BGP config>
```

2.2.16 Multipath

Enables BGP **multipath** on routes belonging to an IPv4 address family. When you do not specify this command, and although BGP has several equivalents, only one path will be installed in the active routing table.

Note

To enable IP multipath routing, you must also specify the type of balancing through the **multipath** command in the IP setup menu.

Multipath can only be formed with equivalent routes (same AS path, metric, etc.). See *Selecting a route* on page 6 for further information.

Syntax:

```
BGP config>multipath [eibgp]
```

eibgp	Enable **multipath** to use iBGP and eBGP paths together. Otherwise, a route received from an external peer (eBGP) is preferred over a route received from an internal peer (iBGP).

Example:

```
BGP config>multipath
BGP config>exit
Config>protocol ip
-- Internet protocol user configuration --
IP config>multipath per-packet
IP config>
```

Command history:

Release	Modification
10.09.25	The **multipath eibgp** option has been introduced.
11.00.04	The **multipath eibgp** option has been introduced.

2.2.17 No aggregate

Removes the configuration of an **aggregate** command for an IPv4 address family.

Note

Since this command applies to routes belonging to an IPv4 address family, we recommend configuring it through its equivalent command in the address-family ipv4 menu.

Syntax:

The syntax is identical to that of the **aggregate** command. The parameters must match those of a previously configured **aggregate** command.

Example:

```
BGP config>no aggregate 10.0.0.0 mask 255.0.0.0 10.0.0.0 mask 255.0.0.0 refines
BGP config>
```

2.2.18 No as

Removes configuration for an AS number where the router is located.

Syntax:

```
BGP config>no as
```

Example:

```
BGP config>no as
BGP config>
```

2.2.19 No as-path

Removes an AS-path from the configuration.

Syntax:

```
BGP config>no as-path <name>
```

name	AS-path name, as defined through the **aspath** command.

Example:

```
BGP config>no as-path "myaspath"
BGP config>
```

2.2.20 No as-path-set

Removes configuration for the **as-path-set** command. If you only specify the AS-path-set name (*name* parameter), all configuration for that AS-path-set will be eliminated. If you also specify the name of another AS-path or AS-path-set (*ref* parameter), that reference will be eliminated from the AS-path-set list.

Syntax:

```
BGP config>no aspath-set <name> [<ref>]
```

name	AS-path-set name.
ref	Name of another defined AS-path or AS-path-set.

Example:

```
BGP config>no as-path-set "myaspathset"
BGP config>
```

2.2.21 No bgp

Removes all bgp configuration.

Syntax:

```
BGP config>no bgp
```

Example:

```
BGP config>no bgp
BGP config>
```

2.2.22 No default-metric

Removes the **default-metric** command configuration.

Syntax:

```
BGP config>no default-metric
```

Example:

```
BGP config>no default-metric
BGP config>
```

2.2.23 No export

Removes an **export** command configuration for an IPv4 address family.

Note

Since this command applies to routes belonging to an IPv4 address family, we recommend configuring it through its equivalent command in the address-family ipv4 menu.

Syntax:

The syntax is identical to that of the **export** command. The parameters must match those of a previously configured **export** command.

Example:

```
BGP config>no export as 200 192.168.0.0 mask 255.255.0.0
BGP config>
```

2.2.24 No generate

Deletes a **generate** command configuration for an IPv4 address family.

Note

Since this command applies to routes belonging to an IPv4 address family, we recommend configuring it through its equivalent command in the address-family ipv4 menu.

Syntax:

The syntax is identical to that of the **generate** command. The parameters must match those of a previously configured **generate** command.

Example:

```
BGP config>no generate default 10.0.0.0 mask 255.0.0.0
BGP config>
```

2.2.25 No graceful-restart

Deletes a **graceful-restart** command configuration.

Syntax:

The syntax is identical to that of the **graceful-restart** command. The parameters must match those of a previously configured graceful-restart command.

Example:

```
BGP config>no graceful-restart enable
BGP config>
```

2.2.26 No group

Deletes a **group** command configuration.

Syntax:

The syntax is identical to that of the **group** command. The parameters must match those of a previously configured **group** command.

Example:

```
BGP config>no group type external peeras 200
BGP config>
```

2.2.27 No import

Deletes an **import** command configuration for an IPv4 address family.

Note

Since this command applies to routes belonging to an IPv4 address family, we recommend configuring it through its equivalent command in the address-family ipv4 menu.

Syntax:

The syntax is identical to that of the **import** command. The parameters must match those of a previously configured **import** command.

Example:

```
BGP config>no import as 200 all
BGP config>
```

2.2.28 No martians

Deletes a **martians** command configuration.

Syntax:

The syntax is identical to that of the **martians** command. The parameters must match those of a previously configured **martians** command.

Example:

```
BGP config>no martians 0.0.0.0 mask 255.0.0.0
BGP config>
```

2.2.29 No multipath

Disables BGP **multipath**.

Syntax:

```
BGP config>no multipath
```

Example:

```
BGP config>no multipath
BGP config>
```

2.2.30 No preference

Removes the preference configuration for routes learned through BGP.

Syntax:

```
BGP config>no preference
```

Example:

```
BGP config>no preference
BGP config>
```

2.2.31 Preference

Defines a preference (**pref)** for routes learned through BGP.

Syntax:

```
BGP config>preference <pref>
```

preference	Specifies a **pref** to assign to learned routes. This preference value can be discarded in favor of a value configured with the **import** or **peer** command.

Example:

```
BGP config>preference 55
BGP config>
```

2.2.32 Vrf

Accesses a specified VRF BGP configuration. On accessing a VRF BGP configuration menu, the prompt changes to *BGP vrf config>*. Use the **exit** command to return to the main VRF BGP configuration.

Syntax:

```
BGP config>vrf <name>
```

name	VRF name where you wish to configure BGP.

Example:

```
BGP config>vrf client1
BGP vrf config>
```

2.2.33 Exit

Finalizes BGP configuration.

Syntax:

```
BGP config>exit
```

Example:

```
BGP config>exit
Config>
```

Chapter 3 BGP Protocol Monitoring

3.1 Monitoring Tools

BGP has the following monitoring mechanisms:

(1) The BGP protocol menu within the router monitoring process.

(2) The BGP events subsystem.

You can obtain detailed information on BGP operations in the router through the BGP events subsystem. For detailed information on protocol events, please see the *els.rtf* events documentation included in the software package.

The BGP monitoring menu can be accessed through the following commands:

```
*monitor
Console Operator
+protocol bgp
BGP+
```

The monitoring menu contains different commands and displays information on the BGP feature. Commands are numbered and come with brief explanations on the information displayed, together with some practical examples.

Note

The BGP monitoring menu commands will not function if BGP is disabled or if it is not supported by the router firmware. If this is the case, a *BGP Unavailable* message will be returned whenever a command is executed.

3.2 Monitoring Commands

3.2.1 Aspaths

Lists information on all autonomous systems that a specific route passes through to reach a destination.

For example, if a route originates in autonomous system 65001, passes through autonomous system 65002 and this, in turn, resends the route to autonomous system 65003, you can say that the autonomous system path, or *AS-path*, is made up of 65001, 65002 and 65003.

The following displays the command output for a router belonging to AS 65002 installed in a similar scenario to the one described above:

```
BGP+aspaths
Ref   Path
0     IGP (Id 1)
0     (65001) 65002 65003 IGP (Id 3)
0     (65001) 65002 IGP (Id 2)
BGP+
```

As you can see, the router is aware of three autonomous system paths.

- The first is the direct path to its IGP.
- The second (with Id 3) is the path originating in AS 65001 (noted in brackets) and passing through 65002 to reach 65003.
- The third (with Id 2) is the path originating in AS 65001 and reaching 65002, which in turn resends the routes to the routers in its AS through the corresponding IGP.

This command gives you a clear idea of the logical topology of the autonomous systems in a network so you know which routes have been learned, from who and through which autonomous systems they have passed before reaching us.

3.2.2 Interfaces

Displays information on the system interfaces acknowledged by the protocol.

The following displays output from the **interfaces** command for a router with an Ethernet interface and a Frame-Relay:

```
BGP+interfaces
#ind name           address         mtu       flags
#6   ethernet0/0.101 192.168.1.101   1500/1436 Up Broadcast Multicast
#6   ethernet0/0.101 2001:db8:1::101 1500/1460 Up Broadcast Multicast
#6   ethernet0/0.101 fe80::2a0:26ff:fe01:ca78 1500/1460 Up Broadcast Multicast
BGP+
```

3.2.3 Memory

Provides detailed information on BGP memory usage in the router.

The tasks shaping the implementation of a protocol generally reserve memory in blocks of the same size. Taking advantage of this characteristic, our implementation of BGP aims to optimize the protocol memory reserves using a pre-reservation policy in the following way:

> **Note**
>
> Each time an X amount of memory is requested from the system, a block is reserved for a fixed unit with assignment Y. This memory is then divided into Y/X blocks of X number of bytes. Thus, the next time the same X amount of memory is requested, one of the previously portioned blocks is given.

The **memory** command first displays the size of the allocated unit (*allocation size*).

It then lists all the memory blocks reserved for the protocol, with the following data displayed in a table:

- *Bck Size:* block size in bytes.
- *Block Name:* an identifier to show the protocol task for which the memory has been reserved.
- *Init:* the number of times this block has initiated.
- *Alloc:* the number of times a block has been requested.
- *Freed:* the number of times a block has been released.
- *In Use:* the number of blocks currently in use.
- *bytes:* bytes currently being used. Calculated by multiplying the *In Use* column by the *Bck Size* column.

Finally, it provides a summary of the total memory reserved for the BGP protocol.

The following displays output from the **memory** command for our router:

```
BGP+memory
Allocation size:  2048
Bck Size Block Name        Init   Alloc  Freed  InUse  (bytes)
   8     asmatch_t         1      2      0      2      (16)
   8     krt_remnant_rt    1      1      1      0      (0)
   8     iflist_t          1      0      0      0      (0)
   8     sockaddr_un.in    1      10143  10101  42     (336)
   12    config_entry      1      0      0      0      (0)
   12    mask_entry        1      162    0      162    (1944)
   12    asp_trans         1      2      0      2      (24)
   12    config_list       1      0      0      0      (0)
   16    dest_mask_intern  1      16     0      16     (256)
   16    task_floating_so  1      1      0      1      (16)
   16    if_primary_list_  1      0      0      0      (0)
   20    trace             1      1      1      0      (0)
   20    bgp_adv_entry     1      21     15     6      (120)
   20    rt_aggr_entry     1      0      0      0      (0)
   20    sradix_node       1      1      0      1      (20)
   20    rt_tsi            1      24     12     12     (240)
   20    radix_node        1      30     16     14     (280)
   20    rt_table          1      2      0      2      (40)
   24    task_block        0      28     0      28     (672)
   24    bgp_rto_entry     1      18     18     0      (0)
   24    bgp_rti_entry     1      0      0      0      (0)
   24    bgp_asp_list      1      18     18     0      (0)
   24    asp_table         1      2      0      2      (48)
   24    asp_state         1      2      0      2      (48)
   24    rt_static_table   1      2      0      2      (48)
   24    rt_walk           1      648    648    0      (0)
   28    task_job          1      3830   3829   1      (28)
   28    krt_delq_entry    1      0      0      0      (0)
   28    if_addr_entry     1      19     0      19     (532)
   28    rt_aggr_head      1      0      0      0      (0)
   36    task_size_block   0      23     0      23     (828)
   36    aux_proto         1      0      0      0      (0)
   36    trace_file        1      1      1      0      (0)
   40    adv_entry         1      31     0      31     (1240)
   40    sockaddr_un.in6   1      20206  20161  45     (1800)
   44    if_info           1      0      0      0      (0)
   60    task_timer        1      14     8      6      (360)
   64    rt_head           1      16     8      8      (512)
   68    gw_entry          1      13     0      13     (884)
   84    as_path           1      14     11     3      (252)
   108   rt_static         1      1      0      1      (108)
   108   krt_q_entry       1      22     22     0      (0)
   116   as_path           1      6      5      1      (116)
   124   bgp_metrics_node  1      9      6      3      (372)
   152   rt_changes        1      0      0      0      (0)
   160   if_link           1      6      0      6      (960)
   208   task              1      16     0      16     (3328)
   512   bgp_asp_hash      1      6      4      2      (1024)
   628   rt_entry          1      17     9      8      (5024)
   1180  if_addr           1      6      0      6      (7080)
   2048  rt_list           1      18     18     0      (0)
   4096  bgp_buffer        1      3      2      1      (4096)
   8168  struct regex_mat  1      0      0      0      (0)
Total Memory: 32652
BGP+
```

3.2.4 Peer-info

The **peer-info** command allows you to view detailed information on BGP peers from a specificautonomous system or peer. The identifier for the autonomous system you want to get information from, or the peer address, are placed into a parameter. The following data is displayed:

- Group type the peer belongs to. This can be:

  Ext: external group.

  Int: internal group.
- Local and remote autonomous system.
- BGP version used.
- Configuration flags enabled.

- Peer state.
- Time lapsed since this peer has been in the established state.
- Metric used to advertise routes to this peer.
- The preference of routes learned from this peer.
- Transmission and reception buffer size.
- Messages sent and received by this peer. This information is broken down in the following way:

  Number of messages sent.

  How many of the sent messages were route *updates* and how many were not.

  Total bytes sent.

  Number of messages received

  How many of the received messages were route *updates* and how many were not.

  Total bytes received.

- Number of times the *established* state has occurred.
- Information on the *Graceful Restart* feature (RFC 4724):

  State: *enabled* or *disabled.*

  Bit policy *Forwarding State* (F): *forced* or *actual*.

  Received *Graceful Restart Capability* message and its content.

  Transmitted *Graceful Restart Capability* message and its content.

  Actual behavior: *restarting speaker* or *receiving speaker*.

  Information on the preserved routes (*stale info*).

  Information on the state of the initial exchange of routes for each family:

  - *received End-of-RIB marker*: indicator signaling the completion of the initial routing update has been received.

  - *sending initial update*: initial routing update has still not finished.

  - *sent End-of-RIB marker*: initial routing update has finished and the indicator to signal completion has been sent.

  - *deferring initial update*: initial routing update is blocked because the route selection process has been delayed.

The following displays the output from the above command for a connected peer belonging to autonomous system 101:

```
BGP+peer-info 233
group type Ext AS 233 local 232 flags <>
 peer 192.168.212.233; version 4; lclInterface ethernet0/0; lcladdr (null); gateway (null)
  flags 0x0
  state 0x6 <Established> (44m36s)
  options 0x100 <VersionSpecified>
  mp options 0x0 <>
  mp address-family 0x3 <ipv4-unicast ipv6-unicast>
  metric_out -1
  preference 170
  preference2 0
  recv buffer size 0
  send buffer size 0
  messages in 354 (updates 18, not updates 336) 7602 octets
  messages out 362 (updates 18, not updates 344) 7842 octets
  changes to "established" 3
  graceful-restart: disabled
    forwarding state (F) bit: actual (set only if preserved)
    received capability: none
    transmitted capability: none
    current role: none
    stale info: none
BGP+
```

3.2.5 Peer-groups

Displays information on the peer groups you have configured. Different peer groups must have different group types. The group type can take the following values:

- *external*: external group.

  internal: internal group.

The following displays the command output for a router configured with three external peers:

```
BGP+peer-groups external
Group    Neighbor        V    AS MsgRcvd MsgSent State
external 172.24.75.22    0  1000       0       0 Connect
external 192.168.43.192  0  2000       0       0 Idle
external 172.24.75.123   4   100       8       8 Established
BGP summary, 3 peers in group type "external"
```

```
BGP+
```

This command is useful when you have multiple peers that belong to different groups. In many user scenarios, the router only communicates with peers that belong to the external group (EBGP). This command produces similar results to the **summary** command.

3.2.6 Reset-message-counters

Resets the meter that counts the number of messages exchanged with all BGP peers. Once the command is executed, said meter (plus any peer updates, non-updates and octets) return to zero.

The console shows the number of BGP peers and groups that have been reset.

The following example displays the command output, followed by a **summary** command and another **peer-info** command showing how the values return to their initial zero value:

```
BGP+reset-message-counters
3 peers in 3 groups reset
BGP+summary
Neighbor        V    AS MsgRcvd MsgSent       State NumEst Time
172.24.75.22    0  1000       0       0     Connect      0 20m13s
192.168.43.192  0  2000       0       0        Idle      0 2m4s
172.24.75.123   4   100       0       0 Established      1 21m18s
BGP summary, 3 groups, 3 peers.
BGP+peer-info 100
group type Ext AS 100 local 200 flags <> peer 172.24.75.123; version 4; lclInterface
(null); lcladdr (null); gateway (null);
  flags 0x0
  state 0x6 <Established> (21m25s)
  options 0x0 <>
  mp options 0x0 <>
  mp address-family 0x1 <ipv4-unicast>
  metric_out -1
  preference 170
  preference2 0
  recv buffer size 0
  send buffer size 0
  messages in 0 (updates 0, not updates 0) 0 octets
  messages out 0 (updates 0, not updates 0) 0 octets
  changes to "established" 1
  graceful-restart: disabled
    forwarding state (F) bit: actual (set only if preserved)
    received capability: none
    transmitted capability: none
    current role: none
    stale info: none
```

If you run the **peer-info** command again after a couple of minutes, you can see if a particular peer is active by checking whether these values have returned to zero. Below, you can see the results of executing a **peer-info** command one minute after executing the **rest-message-counters** command. As you can see, the meter has increased:

```
BGP+peer-info 100
group type Ext AS 100 local 200 flags <> peer 172.24.75.123; version 4; lclInterface
(null); lcladdr (null); gateway (null);
  flags 0x0
  state 0x6 <Established> (9m1s)
  options 0x0 <>
  mp options 0x0 <>
  mp address-family 0x1 <ipv4-unicast>
  metric_out -1
  preference 170
  preference2 0
  recv buffer size 0
  send buffer size 0
  messages in 1 (updates 0, not updates 1) 19 octets
  messages out 1 (updates 0, not updates 1) 19 octets
  changes to "established" 1
  graceful-restart: disabled
    forwarding state (F) bit: actual (set only if preserved)
    received capability: none
    transmitted capability: none
    current role: none
    stale info: none
```

3.2.7 Restart-bgp

Restarts BGP connections. If you specify a peer IP address as the argument, only the connection with that peer is restarted. If you specify *all-peers* as the argument, then all the established connections are restarted.

3.2.8 Routes

Allows BGP-learned routes to be dumped. Its function is similar to the IP monitoring **dump** command or the IPv6 **list routes** command, although there are a few small differences:

- Information is added to an autonomous system path through which a route has been learned if it originates from a BGP peer.
- The format used to display destination networks is slightly different, omitting the host part. For example, network 1.0.0.0 with mask 255.0.0.0 is displayed on the console as *1/8*.
- Tasks handled by BGP are displayed (**preference**, **preference2**, **metric** and **metric2**).
- Other *BGP attributes*, such as communities, are also displayed.

The following example shows output from the command for a router operating with two address families (IPv4 and IPv6). The routes are grouped by family (first IPv4 routes, then IPv6 routes) and may have been learned from different sources.

```
BGP+routes
Flags: A active, M multipath, D deleted, N not install, I incomplete
     Proto        Route/Mask NextHop      Pref Pref2 Metr Metr2  ASPath
A--N- Sta        80.51.73/24  192.168.1.100   60    0   1   0    Incomplete(Id 1)  (a02002)
A--N- Dir       192.168.1/24  192.168.1.102   0     0   1   0    Incomplete(Id 1)  (a00002)
---N- BGP       192.168.1/24  192.168.1.1     170   0   1  none (102)101 Incomplete(Id 2)(202016)
A---- BGP       192.168.2/25  192.168.1.1     170   0   1  none (102)101 Incomplete(Id 2)(a02010)
A---- BGP   192.168.2.128/25  192.168.1.1     170   0   1  none (102)101 Incomplete(Id 2)(a02010)
A---- BGP      192.168.68/24  192.168.1.1     170   0   1  none (102)101 Incomplete(Id 2)(a02010)
A---- BGP     192.168.128/24  192.168.1.1     170   0   1  none (102 101 Incomplete(Id 2)(a02010)
A--N- Dir    2001:db8:1::/64  ethernet0/0.101 0     0   1   0   Incomplete (Id 1)   (a00002)
---N- BGP    2001:db8:1::/64  2001:db8:1::101 170   0   1  none (102)101 Incomplete(Id 3)(202016)
A---- Agg    2001:db8:2::/48  ---             130   0   0   0  (102)101 Incomplete (Id 4)(a06008)
A---- BGP    2001:db8:2::/64  2001:db8:1::101 170   0   0  none(102)101 Incomplete(Id 4)(20a02010)
A---- BGP    2001:db8:5::/64  2001:db8:1::101 170   0   0  none(102)101 Incomplete (Id 5)(a02010)
A--N- Sta 2001:db8:50::20/128 ethernet0/0.101 1     0   0   0   Incomplete(Id 1)   (a00002)
BGP+
```

The flags in the first column do the following:

A: Indicate that the route is active, i.e., it is installed in the corresponding protocol routing table and is being used to route traffic.

M: Indicate this is a multipath route.

D: Indicate the route is being eliminated.

N: Indicate the route will not be installed in the corresponding protocol active routing table. This may be, for example, due to backup routes not being active because the main route is.

I: Incomplete route, i.e., a route whose next hop does not correspond to any network for any interface nor to any other route. Consequently, it cannot be used.

Protocols for those that could have learned the route are as follows:

BGP: route learned from another peer through BGP.

Agg: aggregation route configured in BGP through the **aggregate** command.

Dir: directly connected route installed in the corresponding protocol active routing table.

Sta: static route installed in the corresponding protocol active routing table.

RIP: route learned through RIP or RIPng and installed in the corresponding protocol active routing table.

OSPF: route learned through OSPF or OSFPFv3 and installed in the corresponding protocol active routing table.

The **routes** command dumps the entire contents of the BGP routing table. Since this can amount to a hefty task, there is also the option to dump only a section of the routing tree. This command allows you to filter routes to show by specifying a network address and IPv4 mask. You can also filter IPv6 routes by destination. For example, in the router shown in the above figure, you can show sections of the routing table by adding a parameter to the command:

```
BGP+routes 192.0.0.0/8
Flags: A active, M multipath, D deleted, N not install, I incomplete
     Proto        Route/Mask NextHop      Pref  Pref2  Metr  Metr2  ASPath
A--N- Dir      192.168.1/24  192.168.1.102 0     0     1      0    Incomplete (Id 1) (a00002)
---N- BGP      192.168.1/24  192.168.1.1   170   0     1   none (102) 101 Incomplete (Id 2)(202016)
A---- BGP      192.168.2/25  192.168.1.1   170   0     1   none (102) 101 Incomplete (Id 2)(a02010)
A---- BGP  192.168.2.128/25  192.168.1.1   170   0     1   none (102) 101 Incomplete (Id 2)(a02010)
A---- BGP     192.168.68/24  192.168.1.1   170   0     1   none (102) 101 Incomplete (Id 2)(a02010)
A---- BGP    192.168.128/24  192.168.1.1   170   0     1   none (102) 101 Incomplete (Id 2)(a02010)
BGP+
```

Since routes are grouped independently of their address family in the same table, the **routes** command provides a convenient way to filter them. To display IPv6 routes, you must enter the *ipv6 attribute* together with this command. IPv4 routes can be similarly filtered through the *ipv4 attribute*.

```
BGP+routes ipv6
Flags: A active, M multipath, D deleted, N not install, I incomplete
     Proto         Route/Mask NextHop       Pref  Pref2 Metr  Metr2  ASPath
A--N- Dir   2001:db8:1::/64  ethernet0/0.101  0    0    1    0    Incomplete (Id 1) (a00002)
---N- BGP   2001:db8:1::/64  2001:db8:1::101  170  0    1    none (102)101 Incomplete (Id 3)(202016)
A---- Agg   2001:db8:2::/48  ---              130  0    0    0    (102)101 Incomplete (Id 4)(a06008)
A---- BGP   2001:db8:2::/64  2001:db8:1::101  170  0    0    none (102)101 Incomplete (Id 4)(20a02010)
A---- BGP   2001:db8:5::/64  2001:db8:1::101  170  0    0    none (102)101 Incomplete (Id 5)(a02010)
A--N- Sta 2001:db8:50::20/128 ethernet0/0.101  1   0    0    0    Incomplete (Id 1) (a00002)
BGP+
```

Sometimes the ASPath is so long that only the beginning and end appear, as shown in the following example:

```
BGP+routes
Flags: A active, M multipath, D deleted, N not install, I incomplete
     Proto        Route/Mask NextHop       Pref  Pref2 Metr  Metr2  ASPath
A---- BGP          1.1.1/24 10.0.39.101   170   0     none  none  (1040) 1039 1038 ... 1001
     1000 Incomplete (Id 2)(  a02010)
A--N- Dir         10.0.39/24 10.0.39.100   0     0     1    0     Incomplete (Id 1)   (a00002)
A--N- Dir         10.0.40/24 10.0.40.100   0     0     1    0     Incomplete (Id 1)   (a00002)
BGP+
```

To dump the full ASPath, use the **verbose** option:

```
BGP+ routes verbose
Flags: A active, M multipath, D deleted, N not install, I incomplete
     Proto    Route/Mask NextHop    Pref  Pref2  Metr   Metr2   ASPath
A---- BGP    1.1.1/24 10.0.39.101    170   0     none    none  (1040) 1039 1038 ... 1001 1000
Incomplete (Id 2)(  a02010)
       ASPath: (1040) 1039 1038 1037 1036 1035 1034
               1033 1032 1031 1030 1029 1028 1027 1026
               1025 1024 1023 1022 1021 1020 1019 1018
               1017 1016 1015 1014 1013 1012 1011 1010
               1009 1008 1007 1006 1005 1004 1003 1002
               1001 1000 Incomplete (Id 2)
A--N- Dir    10.0.39/24 10.0.39.100     0     0     1     0  Incomplete (Id 1) (a00002)
A--N- Dir    10.0.40/24 10.0.40.100     0     0     1     0  Incomplete (Id 1) (a00002)
BGP+
```

The following options can be used to filter information:

- **sent_to_peer <peer>**: shows the routes sent to the specified peer.

- **received_from_peer <peer>**: shows the routes received by the specified peer.

```
BGP+routes sent_to_peer 192.168.213.167
Flags: A active, M multipath, D deleted, N not install, I incomplete
     Proto      Route/Mask NextHop         Pref  Pref2 Metr  Metr2  ASPath
A--N- Dir       2.2.2.2/32 2.2.2.2          0     0     1     0     Incomplete (Id 1) (a00002)
A--N- Dir       8.8.8.8/32 8.8.8.8          0     0     1     0     Incomplete (Id 1) (a00002)
A--N- Sta       10.10.10/24 2.2.2.2         60    0     1     0     Incomplete (Id 1) (a02002)
```

3.2.9 Summary

Offers a summary on the general functioning of the BGP protocol.

The following data is displayed:

- Firstly, a line on the general functioning of the protocol. The meaning of the messages in the first line is as follows:

  Configuration running: the read configuration is valid and the protocol is operating.

  Configuration not read: the configuration could not be read.

  Invalid configuration: the configuration read is invalid. This same line will display a message indicating the nature of the problem. The messages can be as follows:

  - *generic error:* an unexpected syntax error has been found on reading the configuration.

  - *too complex:* the autonomous system path is too long. Normally this error occurs when there is undesired recurrence in the configuration.

  - *undefd aspath:* this refers to an autonomous system path that has not been configured.

  - *invalid aspath rep:* a number of repetitions have been configured in an invalid autonomous system path.

  - *out of mem:* the system has run out of memory.

  - *ext config file not found:* this refers to a non-existing file in the configuration.

  - *no local as:* the autonomous system identifier has not been configured in the BGP configuration. BGP cannot operate without this parameter.

  - *no local router-id:* the *router-id* has not been configured in IP protocol configuration. BGP cannot operate without this parameter.

  - *BGP not enabled:* the BGP protocol is not enabled.

  - *config not read:* the configuration cannot be read.

- A full list of the neighbors on whom you have information. The following data is printed for each in table format:

  Neighbor address. This is the IPv4 or IPv6 remote address of the TCP connection.

  Local address. This is the IPv4 or IPv6 local address of the TCP connection.

  BGP version used in the communication: this is 0 if communication has not been established, with 2, 3 or 4 being the BGP versions the router can currently detect.

  The local autonomous system number.

  The remote autonomous system number.

  The number of received and sent messages.

  The current state of the connection, which can be:

  - *Idle*: not attempting to connect to the router. This usually occurs when the peer address is not accessible from the router at a certain time.

  - *Connect*: attempting to actively connect to this peer.

- *Active*: a connection request has been received from this router.

- *OpenSent*: TCP connection with this peer has established and an OPEN packet has been sent to initiate BGP negotiation.

- *OpenConfirm*: expecting confirmation for opening communications with this peer.

- *Established*: connection is now established with this peer.

Time lapsed in the current state.

The number of times the *established* state has occurred.

Time lapsed since this peer has been in the established state.

- Total number of neighbors on whom you have information.
- Total number of configured neighbor groups.

The following example shows the command output for a router configured with two BGP peers:

```
BGP+summary
Configuration running
Neighbor        LclAddr         V LclAS    AS MsgRcvd MsgSent       State StateTime        NumEst EstTime
9.9.9.2         9.9.9.1         4   100   200       8      10 Established 6m37s                 1 6m37s
2222::2332      2222::2331      4   100   200       8      10 Established 6m33s                 1 6m33s
BGP summary, 1 group, 2 peers.
BGP+
```

Command history:

Release	Modification
11.01.09	The local address has been added to the peer summary table as of version 11.01.09.

3.2.10 Tasks

The router uses the notion of cooperative tasks to implement BGP characteristics. You can use the **tasks** command at any time to view the tasks running in the BGP subsystem. The tasks are displayed in table format, with the following information on each:

- *Name:* name of task.
- *Pro:* protocol associated with task.
- *Pri:* task priority, the lower the value shown, the higher the priority.
- *Address:* address associated with task (if this is logical).
- *Port:* port associated with task (if this is logical).
- *S: socket* associated with task (if this is logical).
- *<State-Flags>:* task state and flags.

In the following example, the **tasks** command is executed in one of our routers:

```
BGP+tasks
Name                 Pro Pri Address         Port  S <State - Flags>
IF                   0   10  ---             -1   -1 <Direct - >
INET6                0   15  ---             -1   21 <inet6 - >
INET                 0   15  ---             -1   20 <inet - >
TLDTNET6             0   20  ---             -1   -1 <any - >
TLDTNET              0   20  ---             -1   -1 <any - >
MONIT                0   20  ---             -1   -1 <any - >
Aggregate            0   20  ---             -1   -1 <any - >
RT                   0   20  ---             -1   -1 <any - >
BGPv6                0   40  ::              179  23 <BGP - Accept LowPrio>
BGP                  0   40  0.0.0.0         179  22 <BGP - Accept LowPrio>
BGP_101              0   50  2001:db8:1::101 179  25 <BGP - LowPrio>
BGP_101              0   50  192.168.1.101   179  24 <BGP - LowPrio>
BGP_Group_101_102    0   50  ---             -1   -1 <BGP - LowPrio>
ASPaths              0   50  ---             -1   -1 <any - >
KRT                  255 60  ---             -1   14 <kernel - >
Redirect             0   60  ---             -1   -1 <any - >
```

```
BGP+
```

3.2.11 Timers

There are a number of timers in BGP that might be worth monitoring should a problem occur in our network. The **timers** command is used for this task.

Timers are displayed in table format, with the following information displayed in columns:

- *Name*: name of the timer.
- *Task* : task that created it.
- *Last*: minutes and seconds since it was last executed. If it has never been executed, 00:00 is displayed.
- *Next*: minutes and seconds until an action associated with the timer is executed.
- *Intrvl:* timer execution interval in minutes and seconds.
- *Jitter:* although an action associated with a timer should be executed within a given number of seconds, it might take longer to execute the action if the system is busy with other tasks. The *quality* of the timers is shown in the *Jitter* column. The busier the system is, the greater the amount of jitter in a timer. Ideally, the values in the column will be as close to 00:00 as possible.
- *Flags*: individual characteristics of each timer:

  - If this field is empty, the timer is periodic (an action is executed from time to time).

  - If the field displays *OneShot*, the timer expires once and is then removed from the system.

  - If the field displays *Inactive*, the timer is disabled.

The following displays output from the **timers** command for one of our routers:

```
BGP+timers
Name                  Task     Last   Next   Intrvl Jitter <flags>
AGE                   IF       00:00s 02:47s 00:00s 00:00s <OneShot>
TLDTNET_RT_EXPORT     TLDTNET  00:00s 00:01s 00:01s 00:00s <>
Traffic               BGP_101  00:00s 00:10s 00:00s 00:00s <OneShot>
Traffic               BGP_101  00:00s 00:03s 00:00s 00:00s <OneShot>
IfCheck               KRT      00:13s 00:02s 00:15s 00:00s <>
Timeout               KRT      00:00s 00:00s 00:00s 00:00s <OneShot Inactive>
Age                   Redirect 00:00s 02:47s 00:00s 00:00s <OneShot>
BGP+
```

3.2.12 Vrf

Allows you to access BGP monitoring for a specific VRF. When you access the VRF BGP monitoring menu, the prompt changes to *BGP vrf+*. The **exit** command sends you back to the main VRF BGP monitoring.

Syntax:

```
BGP+vrf <name>
```

name	VRF name where you wish to configure BGP.

Example:

```
BGP+vrf client1
BGP vrf+
```

3.2.13 Exit

Leaves the BGP monitoring menu and returns to general system monitoring. The user is made aware of this when the system displays the + message, as shown below:

Syntax:

```
BGP+exit
```

Example:

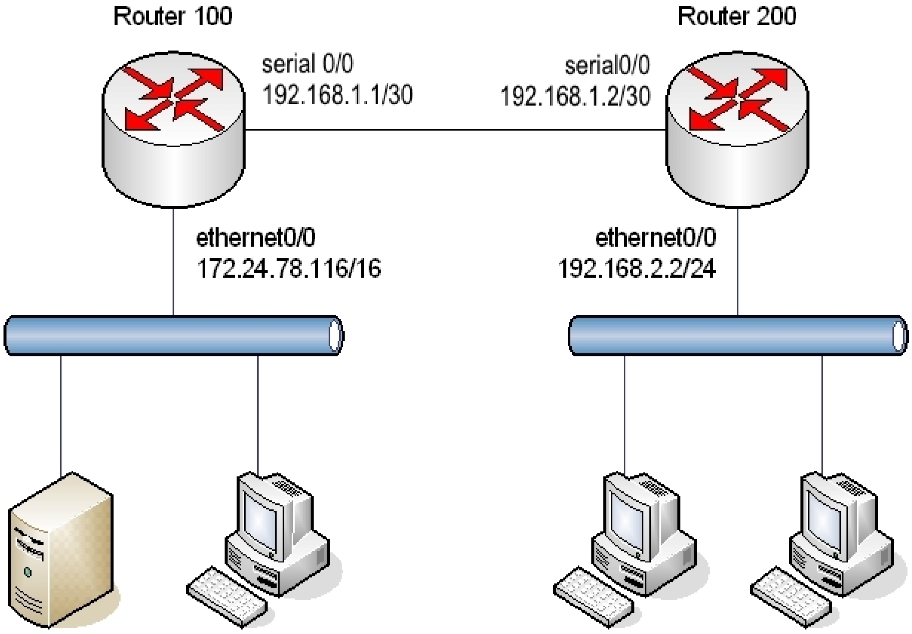
```
BGP+exit
+
```

Chapter 4 Examples

4.1 Basic Example

This example shows a basic eBGP configuration between two of our routers. Only one BGP connection is established between the two autonomous systems over a Frame Relay point-to-point line and both routers export all routes.

The following figure depicts the configured scenario:



Configuration for Router 100 is as follows:

```
no configuration
set hostname Router100
set data-link frame-relay serial0/0
;
network ethernet0/0
; -- Ethernet Interface User Configuration --
   ip address 172.24.78.116 255.255.0.0
exit
;
network serial0/0
; -- Frame Relay user configuration --
   ip address 192.168.1.1 255.255.255.252
;
   pvc 16 default
;
   point-to-point-line 16
   no lmi
exit
;
protocol ip
; -- Internet protocol user configuration --
   router-id 172.24.78.116
exit
;
protocol bgp
; -- Border Gateway Protocol user configuration --
      enable
;
      as 100
;
      address-family ipv4
; -- BGP IPv4 address family configuration --
```

```
         export as 200 prot all all
;
      exit
;
      group type external peer-as 200
; -- BGP group configuration --
         peer 192.168.1.2
      exit
;
exit
```

The configuration for Router 200 is as follows:

```
no configuration
set hostname Router200
set data-link frame-relay serial0/0
;
network ethernet0/0
; -- Ethernet Interface User Configuration --
   ip address 192.168.2.2 255.255.255.0
exit
;
network serial0/0
; -- Frame Relay user configuration --
   ip address 192.168.1.2 255.255.255.252
;
   pvc 16 default
;
   point-to-point-line 16
   no lmi
exit
;
protocol ip
; -- Internet protocol user configuration --
   router-id 192.168.2.2
exit
;
protocol bgp
; -- Border Gateway Protocol user configuration --
      enable
;
      as 200
;
      address-family ipv4
; -- BGP IPv4 address family configuration --
         export as 100 prot all all
;
      exit
;
      group type external peer-as 100
; -- BGP group configuration --
         peer 192.168.1.1
      exit
;
exit
```

The following sections show the monitoring of both routers.

4.1.1 Monitoring Router 100

Access the BGP protocol monitoring menu and check the configuration is correct and a connection is established:

```
Router100 *monitor
Console Operator
Router100 +protocol bgp
Router100 BGP+summary
Configuration running
Neighbor        V    AS MsgRcvd MsgSent NumEst       State Time
```

```
192.168.1.2     4   200      19      20      1 Established 24m13s
BGP summary, 1 group, 1 peer.

Router100 BGP+
```

Check the BGP routing table:

```
Router100 BGP+routes
Flags: A active, M multipath, D deleted, N not install, I incomplete
     Proto  Route/Mask NextHop        Pref Pref2 Metr Metr2  ASPath
A--N- Dir     172.24/16 172.24.78.116  0    0     1    0     Incomplete (Id 1)    (  a00002)
----- dir     172.24/16 172.24.78.116  0    0     1    0     Incomplete (Id 1)    (  200808)
A--N- Dir  192.168.1/30 192.168.1.1    0    0     1    0     Incomplete (Id 1)    (  a00002)
----- dir  192.168.1/30 192.168.1.1    0    0     1    0     Incomplete (Id 1)    (  200808)
---N- BGP  192.168.1/30 192.168.1.2  -170  0     none none  (100) 200 Incomplete (Id 2)(202056)
A---- BGP  192.168.2/24 192.168.1.2   170  0     none none  (100) 200 Incomplete (Id 2)(a02010)
Router100 BGP+
```

As you can see, the router has received routes to networks 192.168.1.0/30 and 192.168.2.0/24 through BGP.

Check the active routing table:

```
Router100 BGP+exit
Router100 +protocol ip
Router100 IP+dump
Type                 Dest net/Mask   Cost Age  Next hop(s)

 Dir(0)[1]         172.24.0.0/16  [  0/1 ] 0   ethernet0/0
Sbnt(0)[0]        192.168.1.0/24  [240/1 ] 0   None
 Dir(0)[1]        192.168.1.0/30  [  0/1 ] 0   serial0/0
 BGP(0)[0]        192.168.2.0/24  [170/1 ] 0   192.168.1.2 (serial0/0)
Routing table size: 768 nets (64512 bytes), 4 nets known, 4 shown
Router100 IP+
```

Verify the route to network 192.168.2.0/24 has been installed.

4.1.2 Monitoring Router 200

Access the BGP protocol monitoring menu and check the configuration is correct and a connection is established:

```
Router200 *monitor
Console Operator
Router200 +protocol bgp
Router200 BGP+summary
Configuration running
Neighbor        V    AS MsgRcvd MsgSent NumEst       State Time
192.168.1.1     4   100      23      24      1 Established 24m52s
BGP summary, 1 group, 1 peer.
Router200 BGP+
```

Check the BGP routing table:

```
Router200 BGP+routes
Flags: A active, M multipath, D deleted, N not install, I incomplete
     Proto Route/Mask NextHop       Pref Pref2 Metr Metr2  ASPath
A---- BGP    172.24/16 192.168.1.1   170  0      none none  (200) 100 Incomplete (Id 2)(  a02010)
A--N- Dir  192.168.1/30 192.168.1.2   0    0     1    0     Incomplete (Id 1)   (  a00002)
----- dir  192.168.1/30 192.168.1.2   0    0     1    0     Incomplete (Id 1)   (  200808)
---N- BGP  192.168.1/30 192.168.1.1  -170  0     none none  (200) 100 Incomplete (Id 2)(  202056)
A--N- Dir  192.168.2/24 192.168.2.2   0    0     1    0     Incomplete (Id 1)   (  a00002)
----- dir  192.168.2/24 192.168.2.2   0    0     1    0     Incomplete (Id 1)   (  200808)

Router200 BGP+
```

As you can see, the router has received routes to networks 172.24.0.0/16 and 192.168.1.0/30 through BGP.

Check the active routing table:

```
Router200 BGP+exit
Router200 +protocol ip
```

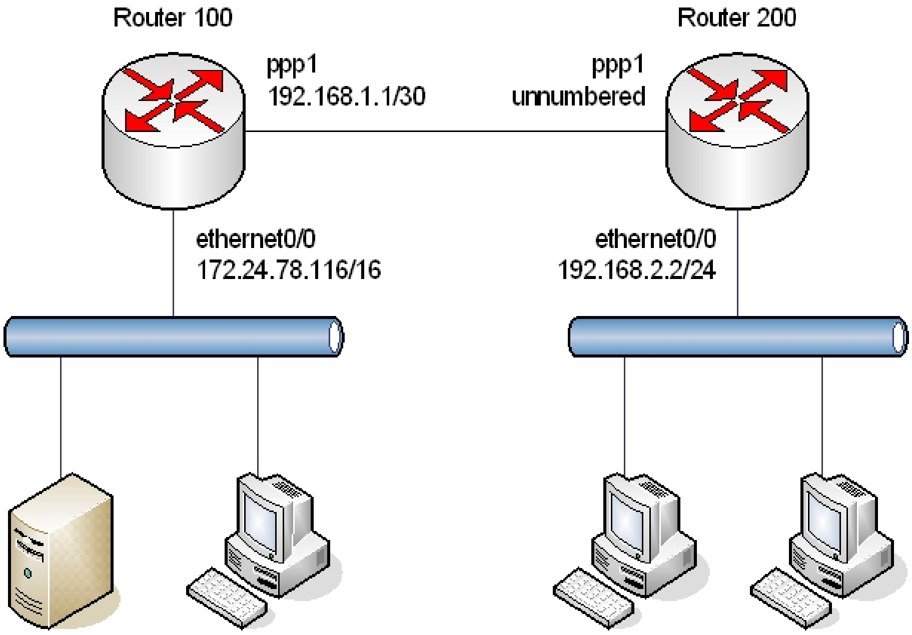
```
Router200 IP+dump
Type                 Dest net/Mask   Cost Age  Next hop(s)

 BGP(0)[0]         172.24.0.0/16  [170/1 ] 0   192.168.1.1 (serial0/0)
Sbnt(0)[0]        192.168.1.0/24  [240/1 ] 0   None
 Dir(0)[1]        192.168.1.0/30  [  0/1 ] 0   serial0/0
 Dir(0)[1]        192.168.2.0/24  [  0/1 ] 0   ethernet0/0

Routing table size: 768 nets (64512 bytes), 4 nets known, 4 shown
Router200 IP+
```

Verify the route to network 172.24.0.0/16 has been installed.

4.2 Example: Address assigned dynamically by the BGP peer

This example shows an eBGP configuration between two of our routers belonging to two autonomous systems connected via BGP over a PPP point-to-point line. One of the routers has an unnumbered address on the PPP interface. In this case, both routers also export all routes.

The following figure depicts the configured scenario:



Configuration for Router 100 is as follows:

```
no configuration
set hostname Router100
add device ppp 1
set data-link sync serial0/0
set data-link x25 serial0/1
set data-link x25 serial0/2

network serial0/0
; -- Interface Synchronous Serial Line. Configuration --
   speed 2048000
   ppp lcp-options mru 2048
exit
;
network ppp1
; -- Generic PPP User Configuration --
   ppp
; -- PPP Configuration --
      authentication pap
      authentication allowed-user myuser ciphered-pwd 0x3861D6897E0D7EA62D381417
D98F3525
      ipcp remote address pool default
   exit
;
   base-interface
```

```
; -- Base Interface Configuration --
      base-interface serial0/0 link
;
   exit
;
exit
;
protocol ip
; -- Internet protocol user configuration --
   router-id 172.24.76.116
;
   address ethernet0/0 172.24.76.116 255.255.0.0
   address ppp1 192.168.1.1 255.255.255.0
;
   pool 192.168.1.100 192.168.1.200
;
exit
;
protocol bgp
; -- Border Gateway Protocol user configuration --
      enable
;
      as 100
;
      address-family ipv4
; -- BGP IPv4 address family configuration --
         export as 200 prot all all
;
      exit
;
      group type external peer-as 200
; -- BGP group configuration --
         allow all
      exit
;
exit
;
dump-command-errors
end
; --- end ---
```

Configuration for Router 200 is as follows:

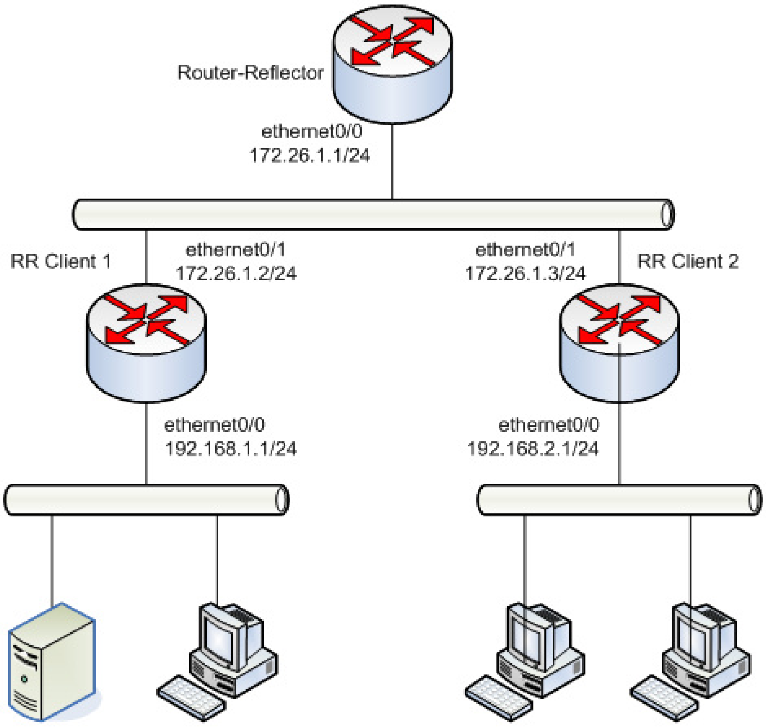
```
no configuration
set hostname Router200
add device ppp 1
set data-link sync serial0/0
set data-link x25 serial0/1
set data-link x25 serial0/2

network serial0/0
; -- Interface Synchronous Serial Line. Configuration --
   speed 2048000
   ppp lcp-options mru 2048
exit
;
network ppp1
; -- Generic PPP User Configuration --
   ppp
; -- PPP Configuration --
      authentication sent-user myuser ciphered-pwd x3861D6897E0D7EA62D381417D98F3525
      ipcp local address assigned
exit
;
   base-interface
; -- Base Interface Configuration --
      base-interface serial0/0 link
;
```

```
   exit
;
exit
;
protocol ip
; -- Internet protocol user configuration --
   router-id 192.168.2.2
;
   address ethernet0/0 192.168.2.2 255.255.255.0
   address ppp1 unnumbered
;
exit
;
protocol bgp
; -- Border Gateway Protocol user configuration --
      enable
;
      as 200
      multipath
;
;
      address-family ipv4
; -- BGP IPv4 address family configuration --
         export as 100 prot all all
;
         import as 100 all
;
      exit
;
      group type external peer-as 100
; -- BGP group configuration --
         peer 192.168.1.1
         peer 192.168.1.1 gateway 192.168.1.1
         peer 192.168.1.1 local-interface ppp1
         peer 192.168.1.1 next-hop-self
         peer 192.168.1.1 no-shared-interface
      exit
;
exit
;
dump-command-errors
end
; --- end ---
```

4.3 Example: Behaving as a route reflector

This example shows an iBGP configuration between three of our routers belonging to the same autonomous system number 100. One of them is acting as a route-reflector for the other two to avoid having to configure a full mesh between the iBGP routers.

The following figure depicts the configured scenario:



Router-reflector configuration is as follows:

```
; Showing Menu and Submenus Configuration for access-level 15 ...
; ATLAS Router 2 156 Version 10.8.0-Alfa

   log-command-errors
   no configuration
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.26.1.1 255.255.0.0
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      router-id 172.26.1.1
;
   exit
;
;
;
   protocol bgp
; -- Border Gateway Protocol user configuration --
      enable
;
      as 100
;
      address-family ipv4
; -- BGP IPv4 address family configuration --
         export as 100 prot all all
;
      exit
;
      group type internal peer-as 100
; -- BGP group configuration --
         peer 172.26.1.2
         peer 172.26.1.2 route-reflector-client
         peer 172.26.1.3
         peer 172.26.1.3 route-reflector-client
      exit
;
   exit
```

```
;
   dump-command-errors
   end
```

Configuration for Client 1 is as follows:

```
  network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.1.1 255.255.255.0
   exit
;
  network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 172.26.1.2 255.255.0.0
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      router-id 172.26.1.2
;
   exit
;
   protocol bgp
; -- Border Gateway Protocol user configuration --
      enable
;
      as 100
;
      address-family ipv4
; -- BGP IPv4 address family configuration --
         export as 100 prot all all
;
      exit
;
      group type internal peer-as 100
; -- BGP group configuration --
         peer 172.26.1.1
      exit
;
   exit
;
   dump-command-errors
   end
```

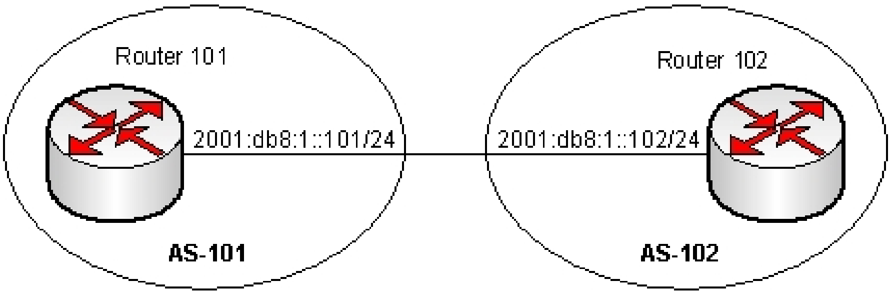
Configuration for Client 2 is as follows:

```
network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.2.1 255.255.255.0
   exit
;
  network ethernet0/1
; -- Ethernet Interface User Configuration –
   ip address 172.26.1.3 255.255.0.0
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      router-id 172.26.1.3
;
   exit
;
   protocol bgp
; -- Border Gateway Protocol user configuration --
      enable
;
      as 100
;
```

```
      address-family ipv4
; -- BGP IPv4 address family configuration --
         export as 100 prot all all
;
      exit
;
      group type internal peer-as 100
; -- BGP group configuration --
         peer 172.26.1.1
      exit
;
   exit
;
   dump-command-errors
   end
```

4.4 Example: Exchanging Routes and IPv6 Connectivity

The following image shows a simple topology made up of a direct connection between two routers, each one belonging to a different autonomous system.



These neighbors use IPv6, both in terms of connectivity and in terms of exchanging routes. Given that we do not want to exchange IPv4 routes, this family is deactivated in the corresponding protocol menu through the **no enable-family** command.

The router in autonomous system 101 exports the default route and all routes learned through BGP using the two **export** rules defined. In addition, this router only installs those routes beginning with the 2001:db8::/48 prefix in the IPv6 routing table. Configuration for this router is as follows:

```
; Showing Menu and Submenus Configuration for access-level 15 ...
; ATLAS Router  6 96 Version 11.00.01
; Warning: dynamic configuration is not saved!

   log-command-errors
   no configuration
   set hostname AS-101
   add device eth-subinterface ethernet0/0 101
   set data-link x25 serial0/1
;
   network ethernet0/0.101
; -- Ethernet Subinterface Configuration --
      ipv6 enable
      ipv6 address 2001:db8:1::101/64
      ipv6 nd ra suppress
      encapsulation dot1q 101
;
;
;
;
   exit
;
   event
; -- ELS Config --
      enable trace subsystem BGP ALL
   exit
;
;
```

```
;
;
;
;
;
   protocol ipv6
; -- IPv6 user configuration --
      route ::/0 interface ethernet0/0.101
      unicast-routing
   exit
;
;
   protocol bgp
; -- Border Gateway Protocol user configuration --
      enable
;
      as 101
      router-id 192.168.1.101
;
      address-family ipv4
; -- BGP IPv4 address family configuration --
         no enable-family
;
      exit
;
      address-family ipv6
; -- BGP IPv6 address family configuration --
         export as 102 prot bgp all
         export as 102 prot static all
;
         import as 102 2001:db8::/48
;
      exit
;
      group type external peer-as 102
; -- BGP group configuration --
         peer 2001:db8:1::102
         peer 2001:db8:1::102 address-family ipv6 unicast
      exit
;
   exit
;
   dump-command-errors
   end
```

Router 102, however, only exports static routes beginning with prefix *2001:db8::/48*, together with all aggregations and all routes learned through BGP. When it comes to importing routes, all routes sent by this protocol will be added to an IPv6 routing table.

The aggregation rule allows a more general route to be created based on a more specific one. Since static route *2001:db8:100:5::/64* meets the aggregation condition (*2001:db8:100::/48 refines*), route *2001:db8:100::/48* is automatically generated.

Finally, the **martians** default rule has been defined because (in this particular case) we do not want to accept the default route.

```
; Showing Menu and Submenus Configuration for access-level 15 ...
; ATLAS Router  7 96 Version 11.00.01
; Warning: dynamic configuration is not saved!

   log-command-errors
   no configuration
   set hostname AS-102
   add device eth-subinterface ethernet0/0 101
;
   network ethernet0/0.101
; -- Ethernet Subinterface Configuration --
      ipv6 address 2001:db8:1::102/64
      encapsulation dot1q 101
```

```
;
;
;
;
   exit
;
   event
; -- ELS Config --
      enable trace subsystem BGP ALL
   exit
;
;
;
;
;
;
   protocol ipv6
; -- IPv6 user configuration --
      route 2001:db8:0:1::/64 interface ethernet0/0.101
      route 2001:db8:0:5::100/128 interface ethernet0/0.101
      route 2001:db8:100:5::/64 interface ethernet0/0.101
      unicast-routing
   exit
;
;
   protocol bgp
; -- Border Gateway Protocol user configuration --
      enable
;
      as 102
      router-id 192.168.1.102
;
      address-family ipv4
; -- BGP IPv4 address family configuration --
         no enable-family
;
      exit
;
      address-family ipv6
; -- BGP IPv6 address family configuration --
         aggregate 2001:db8:100::/48 2001:db8:100::/48 refines
;
         export as 101 prot bgp all
         export as 101 prot static 2001:db8::/48
         export as 101 prot aggregate all
;
         martians default
      exit
;
      group type external peer-as 101
; -- BGP group configuration --
         peer 2001:db8:1::101
         peer 2001:db8:1::101 address-family ipv6 unicast
      exit
;
   exit
;
   dump-command-errors
   end
```