

BGP

- [Summary](#)
- [BGP Terminology](#)
- [BGP Basics](#)
 - [Connection Menu](#)
 - [Session Menu](#)
 - [Template Menu](#)
- [Best-Path Selection](#)
- [Running More than One Instance](#)
 - [VPLS](#)
 - [VPN](#)
 - [Route Distinguisher](#)
 - [Properties](#)

Summary

The Border Gateway Protocol (BGP) allows setting up an inter-domain dynamic routing system that automatically updates routing tables of devices running BGP in case of network topology changes.

BGP is an inter-autonomous system routing protocol based on the distance-vector algorithm. It is used to exchange routing information across the Internet and is the only protocol that is designed to deal with a network of the Internet's size and the only protocol that can deal well with having multiple connections to unrelated routing domains.

BGP is designed to allow for sophisticated administrative routing policies to be implemented. It does not exchange information about network topology but rather a reachability information. As such, BGP is better suited to inter-AS environments and special cases like informational feeds. If you just need to enable dynamic routing in your network, consider OSPF instead.

Standards and Technologies:

- [RFC 4271](#) Border Gateway Protocol 4
- [RFC 4456](#) BGP Route Reflection
- [RFC 5065](#) Autonomous System Confederations for BGP
- [RFC 1997](#) BGP Communities Attribute
- [RFC 8092](#) BGP Large Communities
- [RFC 4360](#), [5668](#) BGP Extended Communities
- [RFC 2385](#) TCP MD5 Authentication for BGPv4
- [RFC 5492](#) Capabilities Advertisement with BGP-4
- [RFC 2918](#) Route Refresh Capability
- [RFC 4760](#) Multiprotocol Extensions for BGP-4
- [RFC 2545](#) Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing
- [RFC 4893](#) BGP Support for Four-octet AS Number Space
- [RFC 4364](#) BGP/MPLS IP Virtual Private Networks (VPNs)
- [RFC 4761](#) Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signalling
- [RFC 6286](#) - AS-wide Unique BGP Identifier for BGP-4

BGP Terminology

- AS - Autonomous System
- ASN - Autonomous System Number
- NLRI - Network Layer Reachability Information is what is being exchanged between BGP peers and represents how to reach the prefixes.
- IGP - Interior Gateway Protocol
- EGP - Exterior Gateway protocol
- RR - Route reflector is the router in the BGP network that reflects advertisements to all the neighbors, avoiding the requirement for full BGP mesh.
- Route server - is the BGP router that does not participate in traffic forwarding. Routes are typically not even installed in the FIB.
- loopback address - a /32 address configured on a dummy bridge interface, that can act as a loopback.

BGP Basics

BGP routers exchange reachability information by means of a transport protocol, which in the case of BGP is TCP (port 179). Upon forming a TCP connection these routers exchange **OPEN** messages to negotiate and confirm supported capabilities.

After agreeing on capabilities to use, the session is considered to be established and peers can start to exchange NLRIs via **UPDATE** messages. This information contains an indication of what sequence of full paths (BGP AS numbers) the route should take in order to reach the destination network (NLRI prefix).

The peers initially exchange their full routing tables and after the initial exchange, incremental updates are sent as the routing tables change. Thus, BGP does not require a periodic refresh of the entire BGP routing table.

BGP maintains the routing table version number which must be the same between any two given peers for the duration of the connection.

KEEPALIVE messages are sent periodically to ensure that the connection is up and running, if **KEEPALIVE** messages are not received within the **Hold Time** interval, the connection will be closed.

To respond to errors or special conditions, **NOTIFICATION** messages can be generated and sent to the remote peer, notification message type also indicates whether the connection should be immediately closed.

There can be two types of BGP connections:

- **iBGP** - is an "internal" link connecting peers from the same AS
- **eBGP** - is an "external" link connecting peers belonging to two different AS-es

A particular AS might have multiple BGP speakers and provide transit service to other AS-es. This implies that BGP speakers must maintain a consistent view of routing within the AS. A consistent view of the routes exterior to the AS is provided by having all BGP routers within the AS establish direct iBGP connections with each other (full mesh) or by utilizing a Router Reflector setup.

Using a set of administrative policies BGP speakers within the AS come to an agreement as to which entry/exit point to use for a particular destination. This information is communicated to the interior routers of the AS using the interior routing protocol (IGP), for example, OSPF, RIP, or static routing. In certain setups, iBGP can take the IGP protocol role as well.

For certain BGP attributes handling behavior may change depending on what type of connection is set up, for example, the LOCAL-PREF attribute is not advertised to eBGP peers.

RouterOS divides configuration and session monitoring into three menus:

- connection menu ([/routing/bgp/connection](#))
- sessions menu([/routing/bgp/session](#))
- template menu ([/routing/bgp/template](#))

Connection Menu

Let's look at a very basic eBGP configuration example assuming, that Router1 IP is 192.168.1.1, AS 65531 and Router2 IP 192.168.1.2, AS 65532:

```
#Router1
/routing/bgp/connection
add name=toR2 remote.address=192.168.1.2 as=65531 local.role=ebgp
```

```
#Router2
/routing/bgp/connection
add name=toR1 remote.address=192.168.1.1 as=65532 local.role=ebgp
```

BGP connection menu defines BGP outgoing connections as well as acts as a template matcher for incoming BGP connections.

`local.role` parameter is used to indicate that this connection will be the eBGP. Also, notice that the connection does not require a remote AS number to be specified, RouterOS can determine a remote AS number dynamically from the first received **OPEN** message.

Parameter equivalent to other vendors and older RouterOS "update-source" is "`local.address`". In most cases, it can be left unconfigured, and let the router determine the address.

When a local address is not specified, BGP will try to guess the local address depending on the current setup:

- if the peer is iBGP

- if loopback available
 - pick the highest loopback address
- if loopback is not available
 - pick any highest IP address on the router
- if the peer is eBGP
 - if a remote peer's IP is not from a directly connected network:
 - and multihop is not set, then throw an error
 - and multihop is enabled:
 - if loopback available
 - pick the highest loopback address
 - if loopback is not available
 - pick any highest IP address on the router
 - if a remote peer's IP is from a directly connected network:
 - and multihop is not set:
 - pick the local routers IP address from that connected network
 - and multihop is set:
 - if loopback available
 - pick the highest loopback address
 - if loopback is not available
 - pick any highest IP address on the router

In addition to connection-specific parameters, template-specific parameters are also directly exposed in this menu, for easier configuration in simple scenarios (when templates are not necessary).

A list of all connection-specific parameters can be seen in the table below:

Property	Description
name (<i>string</i> ; Default:)	Name of the BGP connection
connect (<i>yes / no</i> ; Default: yes)	Whether to allow the router to initiate the connection.
listen (<i>yes / no</i> ; Default: yes)	Whether to listen for incoming connections.
local - a group of parameters associated with the local side of the connection	
.address (<i>IPv4/6</i> ; Default: ::)	Local connection address.
.port (<i>integer [0..65535]</i> ; Default:179)	Local connection port.
.role (<i>ebgp ebgp-customer ebgp-peer ebgp-provider ebgp-rs ebgp-rs-client ibgp ibgp-rr ibgp-rr-client</i> ; Default:)	BGP role, in most common scenarios it should be set to iBGP or eBGP. More information on BGP roles can be found in the corresponding RFC draft https://datatracker.ietf.org/doc/draft-ietf-idr-bgp-open-policy/?include_text=1
.ttl (<i>integer [1..255]</i> ; Default:)	Time To Live (hop limit) that will be recorded in sent TCP packets.
remote - a group of parameters associated with the remote side of the connection	
.address (<i>IPv4/6</i> ; Default: ::)	Remote address used to connect and/or listen to.
.port (<i>integer [0..65535]</i> ; Default:179)	Local connection port.
.as (<i>integer []</i> ; Default:)	Remote AS number. If not specified BGP will determine remote AS automatically from the OPEN message.
.allow-as ()	List of remote AS numbers that are allowed to connect. Useful for dynamic peer configuration.
.ttl (<i>integer [1..255]</i> ; Default:)	Acceptable Time To Live, the hop limit for this TCP connection. For example, if 'ttl=1' then only single-hop neighbors will be able to establish the connection. This property only affects EBGP peers.
tcp-md5-key (<i>string</i> ; Default:)	Key used to authenticate the connection with TCP MD5 signature as described in RFC 2385. If not specified, authentication is not used.
templates (<i>name[,name]</i> ; Default: default)	List of the template names, to inherit parameters from. Useful for dynamic BGP peers.

Session Menu

To see the actual active sessions with selected template parameters and negotiated capabilities refer to the BGP sessions menu:

```
[admin@MikroTik] /routing/bgp/session> print
Flags: E - established
 0 E name="toR2"
  remote.address=192.168.1.2 .as=65532 .id=192.168.1.1 .refused-cap-opt=no
  .capabilities=mp,rr,as4 .afi=ip,ipv6 .messages=43346 .bytes=3635916 .eor=""
  local.address=192.168.1.1 .as=65531 .id=192.168.44.2 .capabilities=mp,rr,gr,as4 .messages=2
  .bytes=71 .eor=""
  output.procid=97 .keep-sent-attributes=no
  .last-notification=ffffffffffffffffffffffffffffffff0015030601
  input.procid=97 .limit-process-routes=500000 ebgp limit-exceeded
  hold-time=3m keepalive-time=1m uptime=4s70ms
```

This menu shows read-only cached BGP session information. It will show the current status of the session, flags, last received notification, and negotiated session parameters.

Even if the BGP session is not active anymore, the cache can still be stored for some time. Routes received from a particular session are removed only if the cache expires, this allows to mitigate extensive routing table recalculations if the BGP session is flapping.

Also, in this menu is located a session-specific set of commands.

Command	Description
clear	Clear the session flags. For example, to be able to re-establish a session after the prefix limit is reached "limit-exceeded" flag must be cleared. It can be done by specifying " flag " parameter, which is able to take the following values: <ul style="list-style-type: none">• input-last-notification• limit-exceeded• output-last-notification• refused-cap-opt• stopped
dump-saved-advertisements	Dump saved advertisements from specified BGP session in the *.pcap file. Filename to store data is set by " save-to " parameter.
refresh	Send route refresh to a specified BGP session. Is used to trigger re-sending all the routes from the remote peer. " address-family " parameters allow specifying for which address family to send route refresh.
resend	Resend prefixes to a specified BGP session. The command takes two parameters: <ul style="list-style-type: none">• "address-family" - parameters allow specifying for which address family to resend prefixes.• "save-to" - name of the pcap file where to dump resent messages, can be used for debugging purposes.
reset	Reset specified BGP session.
stop	Stop specified BGP session.

Template Menu

The template contains all BGP protocol-related configuration options. It can be used as a template for dynamic peers and to apply a similar configuration to a group of peers. Note that this is not the same as peer groups on Cisco devices, where the group is more than just a common configuration.

List of available template parameters:

Property	Description
add-path-out (<i>all /none</i> ; Default:)	
address-families (<i>ip / ipv6 / l2vpn / l2vpn-cisco / vpv4</i> ; Default: <i>ip</i>)	List of address families about which this peer will exchange routing information. The remote peer must support (they usually do) BGP capabilities optional parameter to negotiate any other families than IP.
as (<i>integer [0.. 4294967295]</i> ; Default:)	32-bit BGP autonomous system number. Value can be entered in AS-Plain and AS-Dot formats. The parameter is also used to set up the BGP confederation, in the following format: <i>confederation_as/as</i> . For example, if your AS is 34 and your confederation AS is 43, then as configuration should be <i>as=43/34</i> .
as-override (<i>yes / no</i> ; Default: no)	If set, then all instances of the remote peer's AS number in the BGP AS-PATH attribute are replaced with the local AS number before sending a route update to that peer. Happens before routing filters and prepending.
cisco-vpls-nlri-len-fmt (<i>auto-to-bits / auto-bytes / bits / bytes</i> ; Default:)	VPLS NLRI length format type. Used for compatibility with Cisco VPLS. [[Read more>>]].
cluster-id (<i>IP address</i> ; Default:)	In case this instance is a route reflector: the cluster-ID of the router reflector cluster to this instance belongs. This attribute helps to recognize routing updates that come from another route reflector in this cluster and avoid routing information looping. Note that normally there is only one route reflector in a cluster; in this case, 'cluster-id' does not need to be configured and BGP router ID is used instead
disabled (<i>yes / no</i> ; Default: no)	Whether template is disabled.
hold-time (<i>time[3s..1h] / infinity</i> ; Default: 3m)	Specifies the BGP Hold Time value to use when negotiating with peers. According to the BGP specification, if the router does not receive successive KEEPALIVE and/or UPDATE and/or NOTIFICATION messages within the period specified in the Hold Time field of the OPEN message, then the BGP connection to the peer will be closed. The minimal hold-time value of both peers will be actually used (note that the special value 0 or 'infinity' is lower than any other value) <ul style="list-style-type: none"> infinity - never expire the connection and never send keepalive messages.

input - a group of parameters associated with BGP input

.accept-comunities (<i>string</i> ; Default:)	A quick way to filter incoming updates with specific communities. It allows filtering incoming messages directly before they are even parsed and stored in memory, that way significantly reducing memory usage. Regular input filter chain can only reject prefixes which means that it will still eat memory and will be visible in /routing route table as "not active, filtered". Changes to be applied required session refresh.
.accept-ext-communities (<i>string</i> ; Default:)	A quick way to filter incoming updates with specific extended communities. It allows filtering incoming messages directly before they are even parsed and stored in memory, that way significantly reducing memory usage. Regular input filter chain can only reject prefixes which means that it will still eat memory and will be visible in /routing route table as "not active, filtered". Changes to be applied required session refresh.
.accept-large-communities (<i>string</i> ; Default:)	A quick way to filter incoming updates with specific large communities. It allows filtering incoming messages directly before they are even parsed and stored in memory, that way significantly reducing memory usage. Regular input filter chain can only reject prefixes which means that it will still eat memory and will be visible in /routing route table as "not active, filtered". Changes to be applied required session refresh.
.accept-nlri (<i>string</i> ; Default:)	Name of the ipv4/6 address-list. A quick way to filter incoming updates with specific NLRIs. It allows filtering incoming messages directly before they are even parsed and stored in memory, that way significantly reducing memory usage. Regular input filter chain can only reject prefixes which means that it will still eat memory and will be visible in /routing route table as "not active, filtered". Changes to be applied required session restart.
.accept-unknown (<i>string</i> ; Default:)	A quick way to filter incoming updates with specific "unknown" attributes. It allows filtering incoming messages directly before they are even parsed and stored in memory, that way significantly reducing memory usage. Regular input filter chain can only reject prefixes which means that it will still eat memory and will be visible in /routing route table as "not active, filtered". Changes to be applied required session refresh.

.affinity (afi alone instance main remote-as vrf; Default:)	<p>Configure input multi-core processing. Read more in Routing Protocol Multi-core Support article.</p> <ul style="list-style-type: none"> • alone - input and output of each session are processed in its own process, most likely the best option when there are a lot of cores and a lot of peers • afi, instance, vrf, remote-as - try to run input/output of new session in process with similar parameters • main - run input/output in the main process (could potentially increase performance on single-core even possibly on multi-core devices with small amount of cores) • input - run output in the same process as input (can be set only for output affinity)
.allow-as (<i>integer [0..10]</i> ; Default:)	How many times to allow your own AS number in AS-PATH, before discarding a prefix.
.filter (<i>name</i> ; Default:)	Name of the routing filter chain to be used on input prefixes. Happens after NLRIs are processed. If the chain is not specified, then BGP by default accepts everything.
.ignore-as-path-len (<i>yes / no</i> ; Default: no)	Whether to ignore the AS_PATH attribute in the BGP route selection algorithm
.limit-nlri-diversity (<i>integer</i> ; Default:)	
.limit-process-routes-ipv4 (<i>integer</i> ; Default:)	Try to limit the amount of received IPv4 routes to the specified number. This number does not represent the exact number of routes going to be installed in the routing table by the peer. BGP session "clear" command must be used to reset the flag if the limit is reached.
.limit-process-routes-ipv6 (<i>integer</i> ; Default:)	Try to limit the amount of received IPv6 routes to the specified number. This number does not represent the exact number of routes going to be installed in the routing table by the peer. BGP session "clear" command must be used to reset the flag if the limit is reached.
keepalive-time (<i>time [1s..30m]</i> ; Default: 3m)	How long to keep the BGP session open after the last received "keepalive" message.
multihop (<i>yes / no</i> ; Default: no)	<p>Specifies whether the remote peer is more than one hop away. This option affects outgoing next-hop selection as described in RFC 4271 (for EBGP only, excluding EBGP peers local to the confederation).</p> <p>It also affects:</p> <ul style="list-style-type: none"> • whether to accept connections from peers that are not in the same network (the remote address of the connection is used for this check); • whether to accept incoming routes with NEXT_HOP attribute that is not in the same network as the address used to establish the connection; • the target-scope of the routes installed from this peer; routes from multi-hop or IBGP peers resolve their next-hops through IGP routes by default.
name (<i>string</i> ; Default:)	Name of the BGP template
nexthop-choice (<i>default / force-self / propagate</i> ; Default: default)	<p>Affects the outgoing NEXT_HOP attribute selection. Note that next-hops set in filters always takes precedence. Also note that next-hop is not changed on route reflection, except when it's set in the filter.</p> <ul style="list-style-type: none"> • default - select the next-hop as described in RFC 4271 • force-self - always use a local address of the interface that is used to connect to the peer as the next-hop; • propagate - try to propagate further the next-hop received; i.e. if the route has BGP NEXT_HOP attribute, then use it as the next-hop, otherwise, fall back to the default case
output - a group of parameters associated with BGP output	
.affinity (afi alone instance main remote-as vrf; Default:)	<p>Configure output multicore processing. Read more in Routing Protocol Multi-core Support article.</p> <ul style="list-style-type: none"> • alone - input and output of each session is processed in its own process, the most likely best option when there are a lot of cores and a lot of peers • afi, instance, vrf, remote-as - try to run input/output of new session in process with similar parameters • main - run input/output in the main process (could potentially increase performance on single-core even possibly on multicore devices with small amount of cores) • input - run output in the same process as input (can be set only for output affinity)

.default-originate (<i>always if-installed never</i> ; Default: never)	Specifies default route (0.0.0.0/0) distribution method.
default-prepend (<i>integer [0..255]</i> ; Default:)	
.filter-chain (<i>name</i> ; Default:)	Name of the routing filter chain to be used on the output prefixes. If the chain is not specified, then BGP by default accepts everything.
.filter-select (<i>name</i> ; Default:)	Name of the routing select chain to be used for prefix selection. If not specified, then default selection is used.
.keep-sent-attributes (<i>yes no</i> ; Default: no)	Store in memory sent prefix attributes, required for " dump-saved-advertisements " command to work. By default, sent-out prefixes are not stored to preserve the router's memory. An option should be enabled only for debugging purposes when necessary to see currently advertised prefixes.
.network (<i>name</i> ; Default:)	Name of the address list used to send local networks. The network is sent only if a matching IGP route exists in the routing table.
.no-client-to-client-reflection (<i>yes no</i> ; Default:)	Disable client to client route reflection in Route Reflector setups.
.no-early-cut (<i>yes no</i> ; Default:)	Early cut is the mechanism, to guess (based on default RFC behavior) what would happen with the sent NPLRI when received by the remote peer. If the algorithm determines that the NLRI is going to be dropped, a peer will not even try to send it. However such behavior may not be desired in specific scenarios, then then this option should be used to disable the early cut feature.
redistribute (<i>bgp, connected, copy, dhcp, fantasy, modem, ospf, rip, static, vpn</i> ; Default:)	Enable redistribution of specified route types.
remove-private-as (<i>yes no</i> ; Default: no)	If set, then the BGP AS-PATH attribute is removed before sending out route updates if the attribute contains only private AS numbers. The removal process happens before routing filters are applied and before the local, AS number is prepended to the AS path.
router-id (<i>IP name</i> ; Default: main)	BGP Router ID to be used. Use the ID from the /routing/router-id configuration by specifying the reference name, or set the ID directly by specifying IP. Equal router-ids are also used to group peers into one instance.
routing-table (<i>string</i> ; Default:)	Name of the routing table BGP connections operates on. By default always use the "main" routing table.
save-to (<i>string</i> ; Default:)	Filename to be used to save BGP protocol-specific packet content (Exported PDU) into pcap file. This method allows much simpler peer-specific packet capturing for debugging purposes. Pcap files in this format can also be loaded to create virtual BGP peers to recreate conditions that happened at the time when packet capture was running.
templates (<i>name[,name]</i> ; Default:)	List of template names from which to inherit parameters. Useful feature, to easily configure groups with overlapping configuration options.
use-bfd (<i>yes no</i> ; Default: no)	Whether to use the BFD protocol for faster connection state detection.
vrf (<i>name</i> ; Default: main)	Name of the VRF, to install routes in.

Best-Path Selection

BGP routers can receive multiple copies of the global routing table from multiple providers.

There should be some way to compare those multiple BGP routing tables and select the best route to the destination, solution is the BGP Best Path Selection Algorithm.

The route is evaluated by the algorithm only if it is valid. In general, the route is considered valid if:

- NEXT_HOP of the route is valid and reachable
- AS_PATH received from external peers does not contain the local AS
- the route is not rejected by routing filters

For more information read [nexthop selection and validation](#).

The best path algorithm also compares routes received only by a **single BGP instance**. Routes installed by different BGP instances are compared by the general algorithm, i.e. route distances are compared and the route with lower distance is preferred.

If all the criteria are met, then the following actions take place:

1. The first path received is automatically considered the 'best path'. Any further received paths are compared to the first received to determine if the new path is better.
2. Prefer the path with the highest **WEIGHT**.
This parameter is not a part of the BGP standard, it is invented to quickly locally select the best route. A parameter is local to the router (assigned with routing filters in the BGP input) and cannot be advertised. A route without assigned WEIGHT has a default value of 0.
3. Prefer the path with the highest **LOCAL_PREF**.
This attribute is used only within an AS. A path without the LOCAL_PREF attribute has a value of 100 by default.
4. Prefer the path with the shortest **AS_PATH**. (skipped if `input.ignore-as-path-len` set to **yes**).
Each AS_SET counts as 1, regardless of the set size. The AS_CONFED_SEQUENCE and AS_CONFED_SET are not included in the AS_PATH length.
5. Prefer the path that was locally originated via aggregate or BGP network
6. Prefer the path with the lowest **ORIGIN** type.

Interior Gateway Protocol (IGP) is lower than Exterior Gateway Protocol (EGP), and EGP is lower than INCOMPLETE

in other words **IGP < EGP < INCOMPLETE**

7. Prefer the path with the lowest **multi-exit discriminator** (MED).

The router compares the MED attribute only for paths that have the same neighboring (leftmost) AS. Paths without explicit MED value are treated with MED of 0

8. Prefer **eBGP** over **iBGP** paths
9. Prefer the route that comes from the BGP router with the lowest **router ID**. If a route carries the **ORIGINATOR_ID** attribute, then the **ORIGINATOR_ID** is used instead of the router ID.
10. Prefer the route with the shortest **route reflection cluster list**. Routes without a cluster list are considered to have a cluster list of length 0.
11. Prefer the path that comes from the lowest neighbor address

Running More than One Instance

VPLS

Sub Menu: [/routing/bgp/vpls](#)

This menu lists all the configured bgp-based VPLS instances. These instances allow the router to advertise VPLS BGP NLRI and indicate that the router belongs to a specific customer VPLS network.

MP-BGP based autodiscovery and signaling (RFC 4761).

Cisco VPLS BGP-based auto-discovery (draft-ietf-l2vpn-signaling-08).

Support for multiple import/export route target extended communities for BGP based VPLS (both, RFC 4761 and draft-ietf-l2vpn-signaling-08).

Property	Description
bridge (<i>name</i>)	Name of the bridge where dynamically created VPLS interfaces should be added as ports.
bridge-cost (<i>integer [0..4294967295]</i>)	
bridge-horizon (<i>none integer [0..4294967295]</i>)	If set to none bridge horizon will not be used.
cisco-id ()	Unique identifier. A parameter must be set for cisco-style VPLS signaling. In most cases this should not be used, any modern software supports RFC 4761 style signaling (see <code>site-id</code> parameter). Parameter is a merge of <code>l2-router-id</code> and <code>RD</code> , for example: 10.155.155.1&6550:123
comment (<i>string</i>)	Short description of the item.
disabled (<i>yes no</i>)	Defines whether an item is ignored or used.
export-route-target (<i>list of RTs</i>)	The setting is used to tag BGP NLRI with one or more route targets which on the remote side is used by <code>import-route-targets</code> .
import-route-targets (<i>list of RTs</i>)	The setting is used to determine if BGP NLRI is related to particular VPLS, by comparing route targets received from BGP NLRI.
local-pref (<i>integer[0..4294967295]</i>)	
name (<i>string</i> ; Default:)	
pw-control-word (<i>default disabled enabled</i>)	Enables/disables Control Word usage. Read more in the VPLS Control Word article.
pw-l2mtu (<i>integer [32..65535]</i>)	Advertised pseudowire MTU value.
pw-type (<i>raw-ethernet tagged-ethernet vpls</i>)	The parameter is available starting from v5.16. It allows choosing advertised encapsulation in NLRI used only for comparison. It does not affect the functionality of the tunnel. See pw-type usage example >>

rd (<i>string</i>)	Specifies the value that gets attached to VPLS NLRI so that receiving routers can distinguish advertisements that may otherwise look the same. This implies that a unique route-distinguisher for every VPLS must be used. It is not necessary to use the same route distinguisher for some VPLS on all routers forming that VPLS as distinguisher is not used for determining if some BGP NLRI is related to a particular VPLS (Route Target attribute is used for this), but it is mandatory to have different distinguishers for different VPLSes. Accepts 3 types of formats. Read more>>
site-id (<i>integer [0..65535]</i>)	Unique site identifier. Each site must have a unique site-id. A parameter must be set for RFC 4761 style VPLS signaling.
vrf (<i>name</i>)	Name of the VRF table.

VPN

Sub Menu: </routing/bgp/vpn>

Route Distinguisher

Route Distinguisher is a 64bit integer, which is divided into three parts: type (always 2 bytes), administrator, and value.

Currently, there are three format types defined.

2bytes	2bytes	2bytes	2bytes
Type1	ASN	4byte value	
Type2	4-byte IP		value
Type3	4-byte ASN		value

Properties

Property	Description
disabled (<i>yes / no</i>)	
export-filter (<i>name</i>)	The name of the routing filter chain is used to filter prefixes before exporting.
export-route-targets (<i>rt[,rt]</i>)	Route targets to be added when exporting VPNv4 routes. The accepted RT format is similar to one for Route Distinguishers.
import-filter (<i>integer [0..4294967295]</i>)	The name of the routing filter chain is used to filter prefixes before importing.
import-route-targets (<i>rt[,rt]</i>)	Import VPNv4 routes with matching route targets. The accepted RT format is similar to one for Route Distinguishers.
label-allocation-policy (<i>per-prefix / per-vrf</i>)	
route-distinguisher (string)	Helps to distinguish between overlapping routes from multiple VRFs. Should be unique per VRF. Accepts 3 types of formats. Read more>>
vrf (<i>name</i>)	Name of the VRF table to install routes to.