Manual:IP/Route

From MikroTik Wiki

< Manual:IP

Contents
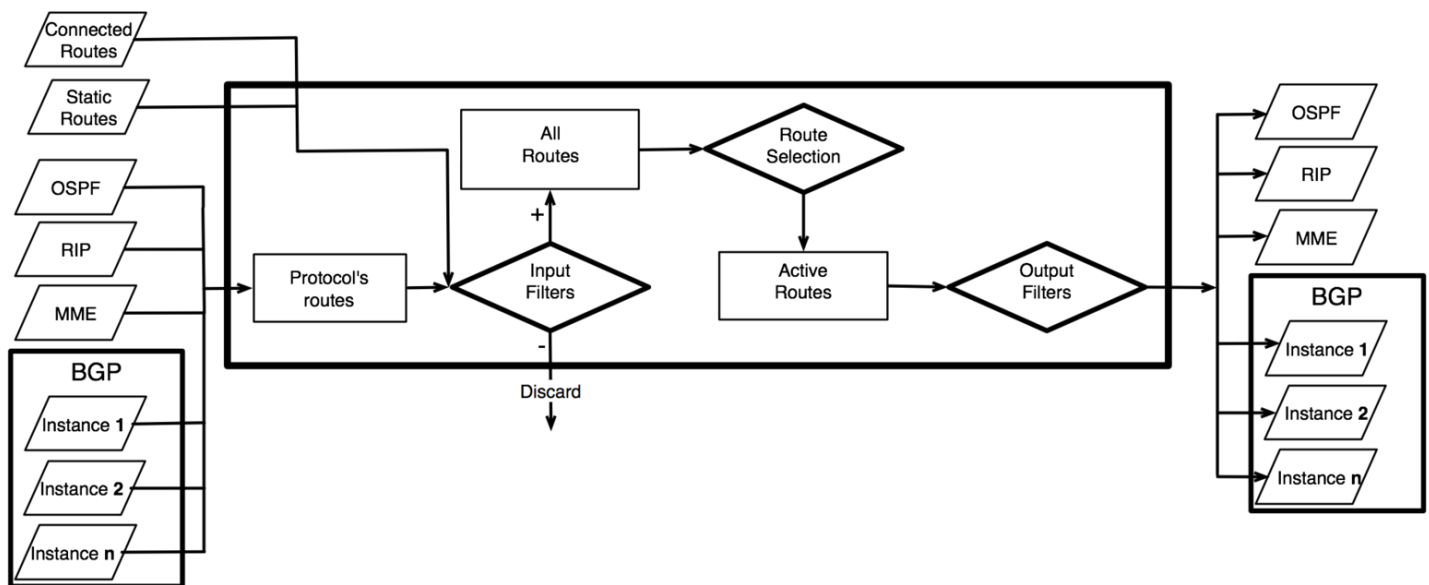
Overview

Router keeps routing information in several separate spaces:

- FIB (Forwarding Information Base), that is used to make packet forwarding decisions. It contains a copy of the necessary routing information.
- Each routing protocol (except BGP) has it's own internal tables. This is where per-protocol routing decisions are made. BGP does not have internal routing tables and stores complete routing information from all peers in the RIB.
- RIB contains routes grouped in separate routing tables based on their value of **routing-mark**. All routes without **routing-mark** are kept in the *main* routing table. These tables are used for best route selection. The *main* table is also used for nexthop lookup.

Routing Information Base

Connected Routes

Static Routes

OSPF

RIP

MME

BGP
Instance 1
Instance 2
Instance n

Protocol's routes

All Routes

Input Filters

Route Selection

Active Routes

Output Filters

+

−

Discard

OSPF

RIP

MME

BGP
Instance 1
Instance 2
Instance n

RIB (Routing Information Base) contains complete routing information, including static routes and policy routing rules configured by the user, routing information learned from routing protocols, information about connected networks. RIB is used to filter routing information, calculate best route for each destination prefix, build and update Forwarding Information Base and to distribute routes between different routing protocols.

By default forwarding decision is based only on the value of destination address. Each route has **dst-address** property, that specifies all destination addresses this route can be used for. If there are several routes that apply to a particular IP address, the most specific one (with largest netmask) is used. This operation (finding the most specific route that matches given address) is called *routing table lookup*.

If routing table contains several routes with the same **dst-address**, only one of them can be used to forward packets. This route is installed into FIB and marked as *active*.

When forwarding decision uses additional information, such as a source address of the packet, it is called *policy routing*. Policy routing is implemented as a list of policy routing rules, that select different routing table based on destination address, source address, source interface, and routing mark (can be changed by firewall mangle rules) of the packet.

All routes by default are kept in the *main* routing table. Routes can be assigned to specific routing table by setting their **routing-mark** property to the name of another routing table. Routing tables are referenced by their name, and are created automatically when they are referenced in the configuration.

Each routing table can have only one active route for each value of **dst-address** IP prefix.

There are different groups of routes, based on their origin and properties.

Default route

Route with **dst-address** *0.0.0.0/0* applies to every destination address. Such route is called the *default route*. If routing table contains an active default route, then routing table lookup in this table will never fail.

Connected routes

Connected routes are created automatically for each IP network that has at least one enabled interface attached to it

(as specifie in the `/ip address` configuration). RIB tracks status of connected routes, but does not modify them. For each connected route there is one *ip address* item such that:

- address part of **dst-address** of connected route is equal to **network** of *ip address* item.
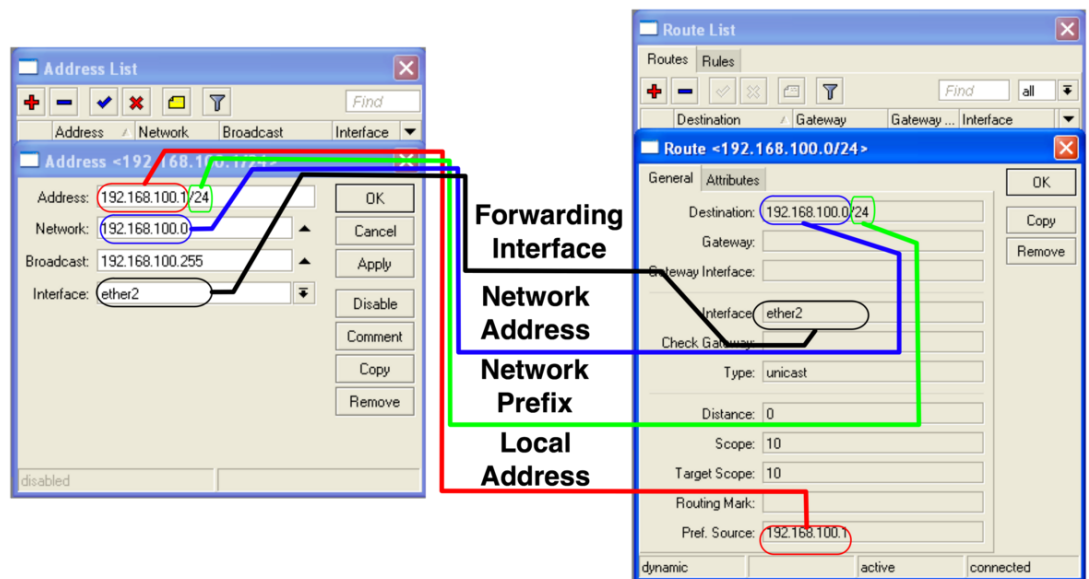
- netmask part of **dst-address** of connected route is equal to netmask part of **address** of *ip address* item.
- **pref-src** of connected route is equal to address part of **address** of *ip address* item.
- **interface** of connected route is equal to **actual-interface** of *ip address* item (same as **interface**, except for bridge interface ports).

Multipath (ECMP) routes

To implement some setups, such as load balancing, it might be necessary to use more than one path to given destination. However, it is not possible to have more than one active route to destination in a single routing table.

ECMP (Equal cost multi-path) routes have multiple **gateway** nexthop values. All reachable nexthops are copied to FIB and used in forwarding packets.

OSPF protocol can create ECMP routes. Such routes can also be created manually.

Because results of the forwarding decision are cached, packets with the same **source address, destination address, source interface, routing mark** and **ToS** are sent to the same gateway. This means that one connection will use only one link in each direction, so ECMP routes can be used to implement per-connection load balancing. See interface bonding if you need to achieve per-packet load balancing.

Routes with interface as a gateway

Value of **gateway** can be specified as an interface name instead of the nexthop IP address. Such route has following special properties:

- Unlike connected routes, routes with interface nexthops are not used for nexthop lookup.
- It is possible to assign several interfaces as a value of **gateway**, and create ECMP route. It is not possible to have connected route with multiple **gateway** values.

Route selection

Each routing table can have one active route for each destination prefix. This route is installed into FIB. Active route is selected from all candidate routes with the same **dst-address** and **routing-mark**, that meet the criteria for becoming an active route. There can be multiple such routes from different

routing protocols and from static configuration. Candidate route with the lowest **distance** becomes an active route. If there is more than one candidate route with the same **distance**, selection of active route is arbitrary (except for BGP routes).

BGP has the most complicated selection process (described in separate article). Notice that this protocol-internal selection is done only after BGP routes are installed in the main routing table; this means there can be one candidate route from each BGP peer. Also note that BGP routes from different BGP instances are compared by their distance, just like other routes.

Criteria for selecting candidate routes

To participate in route selection process, route has to meet following criteria:
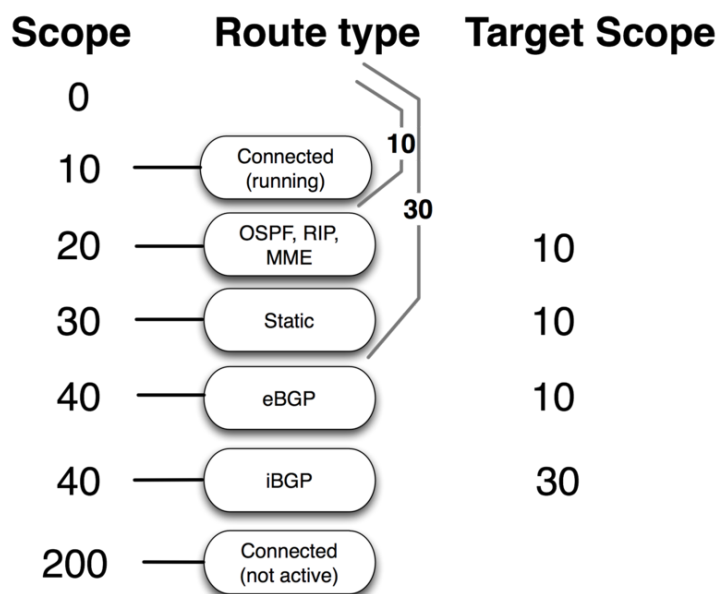
- route is not disabled.
- **distance** is not *255*. Routes that are rejected by route filter have **distance** value of *255*.
- **pref-src** is either not set or is a valid local address of the router.
- **routing-mark** is either not set or is referred by firewall or policy routing rules.
- If type of route is *unicast* and it is not a connected route, it must have at least one reachable nexthop.

Nexthop lookup

Nexthop lookup is a part of the route selection process.

Routes that are installed in the FIB need to have interface associated with each gateway address. Gateway address (nexthop) has to be directly reachable via this interface. Interface that should be used to send out packets to each gateway address is found by doing nexthop lookup.

Some routes (e.g. iBGP) may have gateway address that is several hops away from this router. To install such routes in the FIB, it is necessary to find the address of the directly reachable gateway (an immediate nexthop), that should be used to reach the gateway address of this route. Immediate nextop addresses are also found by doing nexthop lookup.

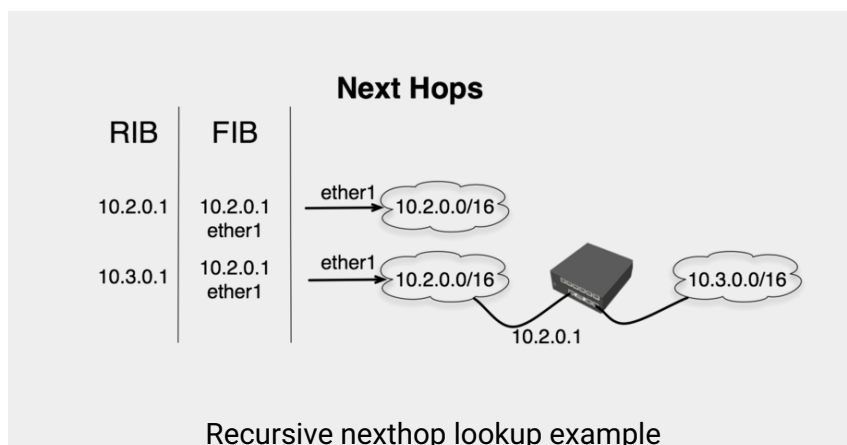| Scope	Route type	Target Scope
0		
10	Connected (running)	
20	OSPF, RIP, MME	10
30	Static	10
40	eBGP	10
40	iBGP	30
200	Connected (not active)	

(annotations near Connected/Static: 10, 30)

Nexthop lookup is done only in the *main* routing table, even for routes with different value of **routing-mark**. It is necessary to restrict set of routes that can be used to look up immediate nexthops. Nexthop values of RIP or OSPF routes, for example, are supposed to be directly reachable and should be looked up only using connected routes. This is achieved using **scope** and **target-scope** properties.

- Routes with interface name as the value of gateway are not used for nexthop lookup. If route has both interface nexthops and active IP address nexthops, then interface nexthops are ignored.
- Routes with **scope** greater than the maximum accepted value are not used for nexthop lookup. Each route specifies maximum accepted **scope** value for it's nexthops in the **target-scope** property. Default value of this property allows nexthop lookup only through connected routes, with the exception of iBGP routes that have larger default value and can lookup nexthop also through IGP and static routes.

Interface and immediate nexthop are selected based on the result of nexthop lookup:

- If most specific active route that nexthop lookup finds is connected route,

Next Hops

RIB	FIB	
10.2.0.1	10.2.0.1 ether1	ether1 → 10.2.0.0/16
10.3.0.1	10.2.0.1 ether1	ether1 → 10.2.0.0/16 ... 10.3.0.0/16

10.2.0.1

Recursive nexthop lookup example

then interface of this connected route is used as the nexthop interface, and this gateway is marked as *reachable*. Since gateway is directly reachable through this interface

- nexthop 10.2.0.1 is resolved through a connected route, it's status is *reachable*.
- nexthop 10.3.0.1 is resolved recursively through a 10.3.0.0/16 route, it's status is *recursive*, and it uses 10.2.0.1 as the immediate nexthop value that is installed in the FIB.

(that's exactly what connected route means), the gateway address is used as the immediate nexthop address.
- If most specific active route that nexthop lookup finds has nexthop that is already resolved, immediate nexthop address and interface is copied from that nexthop and this gateway is marked as *recursive*.
- If most specific active route that nexthop lookup finds is ECMP route, then it uses first gateway of that route that is not *unreachable*.
- If nexthop lookup does not find any route, then this gateway is marked as *unreachable*.

Forwarding Information Base

FIB (Forwarding Information Base) contains copy of information that is necessary for packet forwarding:

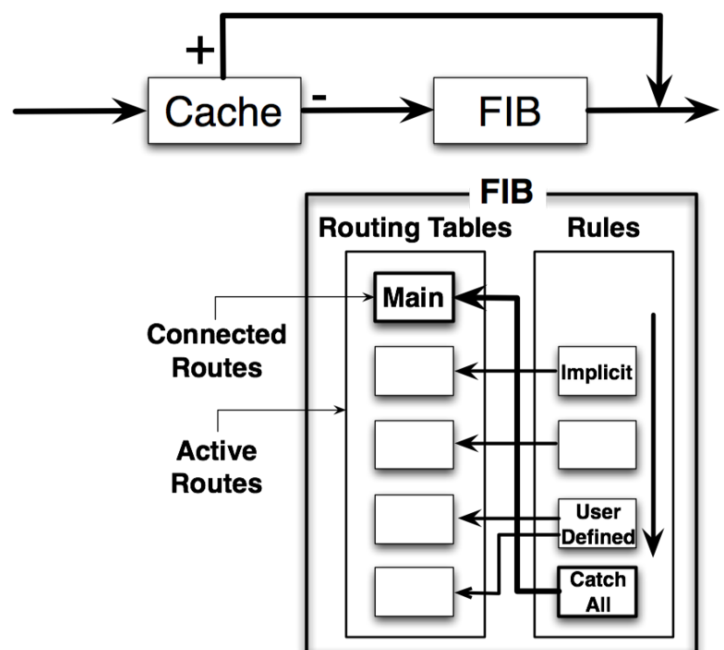- all active routes
- policy routing rules

By default (when no **routing-mark** values are used) all active routes are in the *main* table, and there is only one hidden implicit rule ("catch all" rule) that uses the *main* table for all destination lookups.

Routing table lookup

FIB uses following information from packet to determine it's destination:

- source address

- destination address
- source interface
- routing mark
- ToS (not used by RouterOS in policy routing rules, but it is a part of routing cache lookup key)

Possible routing decisions are:

- receive packet locally
- discard packet (either silently or by sending ICMP message to the sender of the packet)
- send packet to specific IP address on specific interface

Results of routing decision are remembered in the routing cache. This is done to improve forwarding performance. When another packet with the same **source address, destination address, source interface, routing mark** and **ToS** is routed, cached results are used. This also allows to implement per-connection load balancing using ECMP routes, because values used to lookup entry in the routing cache are the same for all packets that belong to the same connection and go in the same direction.

If there is no routing cache entry for this packet, it is created by running routing decision:

- check that packet has to be locally delivered (destination address is address of the router)
- process implicit policy routing rules
- process policy routing rules added by user
- process implicit catch-all rule that looks up destination in the *main* routing table
- return result is "network unreachable"

Rules that do not match current packet are ignored. If rule has action *drop* or *unreachable*, then it is returned as a result of the routing decision process. If action is *lookup* then destination address of the packet is looked

Result of routing decision can be:

- IP address of nexthop + interface
- point-to-point

up in routing table that is specified in the rule. If lookup fails (there is no route that matches destination address of packet), then FIB proceeds to the next rule. Otherwise:

- interface
- local delivery
- discard
- ICMP prohibited
- ICMP host unreachable
- ICMP network unreachable

- if type of the route is *blackhole*, *prohibit* or *unreachable*, then return this action as the routing decision result;
- if this is a connected route, or route with an interface as the **gateway** value, then return this interface and the destination address of the packet as the routing decision result;
- if this route has IP address as the value of **gateway**, then return this address and associated interface as the routing decision result;
- if this route has multiple values of nexthop, then pick one of them in round robin fashion.

Result of this routing decision is stored in new routing cache entry.

Properties

Route flags

Property(Flag)	Description
`disabled` (*X*)	Configuration item is disabled. It does not have any effect on other routes and is not used by forwarding or routing protocols in any way.
`active` (*A*)	Route is used for packet forwarding. See

	route selection.
dynamic (*D*)	Configuration item created by software, not by management interface. It is not exported, and cannot be directly modified.
connect (*C*)	connected route.
static (*S*)	static route.
rip (*r*)	RIP route.
bgp (*b*)	BGP route.
ospf (*o*)	OSPF route.
mme (*m*)	MME route.
blackhole (*B*)	Silently discard packet forwarded by this route.
unreachable (*U*)	Discard packet forwarded by this route. Notify sender with ICMP host

	unreachable (type 3 code 1) message.
prohibit (*P*)	Discard packet forwarded by this route. Notify sender with ICMP communication administratively prohibited (type 3 code 13) message.

General properties

Property	Description
check-gateway (*arp* | *ping*; Default: "")	Periodically (every 10 seconds) check gateway by sending either ICMP echo request (*ping*) or ARP request (*arp*). If no response from gateway is received for 10 seconds, request times

	out. After two timeouts gateway is considered unreachable. After receiving reply from gateway it is considered reachable and timeout counter is reset.
comment (*string*; Default: "")	Description of particular route

distance (*integer[1..255]*; Default: **"1"**)	Value used in route selection. Routes with smaller distance value are given preference. If value of this property is not set, then the default depends on route protocol:
	
	  • connected routes: 0
	  • static routes: 1
	  • eBGP: 20
	  • OSPF: 110
	  • RIP: 120
	  • MME: 130
	  • iBGP: 200

dst-address (*IP prefix*; Default: **0.0.0.0/0**)	IP prefix of route, specifies destination addresses that this route can be used for. Netmask part of this property specifies how many of the most

	significant bits in packet destination address must match this value. If there are several active routes that match destination address of packet, then the most specific one (with largest netmask value) is used.
gateway (*IP | interface | IP%interface | IP@table[, IP | string, [..]]*; Default: "")	Array of IP addresses or interface names. Specifies which host or interface packets should be sent to. Connected routes and routes with *blackhole, unreachable* or *prohibit* type do not have this property. Usually value of this property is a

single IP address of a gateway that can be directly reached through one of router's interfaces (but see nexthop lookup). ECMP routes have more than one gateway value. Value can be repeated several times.

`pref-src` (*IP*; Default: "")	Which of the local IP addresses to use for locally originated packets that are sent via this route. Value of this property has no effect on forwarded packets. If value of this property is set to IP address that is not local address of this router then the route

	will be inactive. If **pref-src** value is not set, then for locally originated packets that are sent using this route router will choose one of local addresses attached to the output interface that match destination prefix of the route (an example).
route-tag (*integer*; Default: "")	Value of route tag attribute for RIP or OSPF. For RIP only values 0..4294967295 are valid.
routing-mark (*string*; Default: "")	Name of routing table that contains this route. Not set by default which is the same as *main*. Packets that are

	marked by firewall with this value of **routing-mark** will be routed using routes from this table, unless overridden by policy routing rules. Not more than 251 routing marks can be added per router.

`scope` (*integer[0..255]*; Default: **"30"**)	Used in nexthop resolution. Route can resolve nexthop only through routes that have **scope** less than or equal to the **target-scope** of this route. Default value depends on route protocol:
	
	  - connected routes: 10 (if interface is running)
	  - OSPF, RIP, MME routes: 20
	  - static routes: 30
	  - BGP routes: 40
	  - connected routes: 200 (if interface is not running)

`target-scope` (*integer[0..255]*; Default: **"10"**)	Used in nexthop resolution. This is the maximum value of **scope** for a route through which a nexthop of this route can be resolved. See nexthop lookup. For

	iBGP value is set to 30 by default.
type (*unicast | blackhole | prohibit | unreachabl*; Default: **unicast**)	Routes that do not specify nexthop for packets, but instead perform some other action on packets have type different from the usual *unicast*. *blackhole* route silently discards packets, while *unreachable* and *prohibit* routes send ICMP Destination Unreachable message (code 1 and 13 respectively) to the source address of the packet.
vrf-interface (*string*; Default: **"10"**)	VRF interface name

Other Read-only properties

Property	Description
gateway-status (*array*)	Array of gateways, gateway states and which interface is used for forwarding. Syntax "IP state interface", for example "10.5.101.1 reachable bypass-bridge". State can be *unreachable, reachable* or *recursive*. See nexthop lookup for details.
ospf-metric (*integer*)	Used OSPF metric for particular route
ospf-type (*string*)	

BGP Route Properties

These properties contain information that is used by BGP routing protocol. However, values of these properties can be set for any type of route, including static and connected. It can be done either manually (for static routes) or using route filters.

Property	Description

bgp-as-path (*string*; Default: "")	Value of BGP AS_PATH attribute. Comma separated list of AS numbers with confederation AS numbers enclosed in *()* and AS_SETs enclosed in *{}*. Used to check for AS loops and in BGP route selection algorithm: routes with shorter AS_PATH are preferred (but read how AS_PATH length is calculated).
bgp-atomic-aggregate (*yes | no*; Default:)	Value of BGP ATOMIC_AGGREGATE attribute.
bgp-communities (*array of (integer:integer | internet | no-advertise | no-export |local-as*; Default:)	Value of BGP communities list. This attribute can be used to group or filter routes. Named values have special meanings: • *internet* - advertise this route to the Internet community (i.e. all routers) • *no-advertise* - do not advertise this route to any peers • *no-export* - do not advertise this route to EBGP peers • *local-as* - same as *no-export*, except that route is also advertised to EBGP peers inside local confederation

bgp-local-pref (*integer*; Default:)	Value of BGP LOCAL_PREF attribute. Used in BGP route selection algorithm: routes with greater LOCAL_PREF value are preferred. If value is not set then it is interpreted as 100.
bgp-med (*integer*; Default:)	Value of BGP MULTI_EXIT_DISC BGP attribute. Used in BGP route selection algorithm: routes with lower MULTI_EXIT_DISC value are preferred.. If value is not set then it is interpreted as 0.
bgp-origin (*igp* | *egp* | *incomplete*; Default:)	Value of BGP ORIGIN attribute. Used in BGP

	route selection algorithm: *igp* routes are preferred over *egp* and *egp* over *incomplete*.
bgp-prepend (*integer [0..16]*; Default:)	How many times to prepend router's own AS number to AS_PATH attribute when announcing route via BGP. Affects only routes sent to eBGP peers (for iBGP value 0 is always used).

Read-only

Property	Description
bgp-ext-communities (*string*)	Value of BGP extended communities attribute
bgp-weight (*integer*)	Additional value used by BGP best path selection algorithm.

	Routes with higher weight are preferred. It can be set by incoming routing filters and is useful only for BGP routes. If value is not set then it is interpreted as 0.
received-from (*string*)	Name of the BGP peer from which route is received.

- This page was last edited on 6 December 2017, at 14:49.