```
# other daemons...
bfdd_options="  --daemon -A 127.0.0.1 -M grpc"
```

2.4 Filtering

FRR provides many very flexible filtering features. Filtering is used for both input and output of the routing information. Once filtering is defined, it can be applied in any direction.

2.4.1 IP Access List

access-list NAME [seq (1-4294967295)] permit IPV4-NETWORK

access-list NAME [seq (1-4294967295)] deny IPV4-NETWORK

seq seq *number* can be set either automatically or manually. In the case that sequential numbers are set manually, the user may pick any number less than 4294967295. In the case that sequential number are set automatically, the sequential number will increase by a unit of five (5) per list. If a list with no specified sequential number is created after a list with a specified sequential number, the list will automatically pick the next multiple of five (5) as the list number. For example, if a list with number 2 already exists and a new list with no specified number is created, the next list will be numbered 5. If lists 2 and 7 already exist and a new list with no specified number is created, the new list will be numbered 10.

Basic filtering is done by *access-list* as shown in the following example.

```
access-list filter deny 10.0.0.0/9
access-list filter permit 10.0.0.0/8
access-list filter seq 13 permit 10.0.0.0/7
```

show <ip|ipv6> access-list [json]

Display all IPv4 or IPv6 access lists.

If the `json` option is specified, output is displayed in JSON format.

show <ip|ipv6> access-list WORD [json]

Display the specified IPv4 or IPv6 access list.

If the `json` option is specified, output is displayed in JSON format.

2.4.2 IP Prefix List

ip prefix-list provides the most powerful prefix based filtering mechanism. In addition to *access-list* functionality, *ip prefix-list* has prefix length range specification and sequential number specification. You can add or delete prefix based filters to arbitrary points of prefix-list using sequential number specification.

If no ip prefix-list is specified, it acts as permit. If *ip prefix-list* is defined, and no match is found, default deny is applied.

ip prefix-list NAME (permit|deny) PREFIX [le LEN] [ge LEN]

ip prefix-list NAME seq NUMBER (permit|deny) PREFIX [le LEN] [ge LEN]

You can create *ip prefix-list* using above commands.

seq seq *number* can be set either automatically or manually. In the case that sequential numbers are set manually, the user may pick any number less than 4294967295. In the case that sequential number are set automatically, the sequential number will increase by a unit of five (5) per list. If a list with no specified sequential number is created after a list with a specified sequential number, the list will automatically pick the next multiple of five (5) as the list number. For example, if a list with number 2 already exists and a new list with no specified number is created, the next list will be numbered 5. If lists 2 and 7 already exist and a new list with no specified number is created, the new list will be numbered 10.

le Specifies prefix length. The prefix list will be applied if the prefix length is less than or equal to the le prefix length.

ge Specifies prefix length. The prefix list will be applied if the prefix length is greater than or equal to the ge prefix length.

Less than or equal to prefix numbers and greater than or equal to prefix numbers can be used together. The order of the le and ge commands does not matter.

If a prefix list with a different sequential number but with the exact same rules as a previous list is created, an error will result. However, in the case that the sequential number and the rules are exactly similar, no error will result.

If a list with the same sequential number as a previous list is created, the new list will overwrite the old list.

Matching of IP Prefix is performed from the smaller sequential number to the larger. The matching will stop once any rule has been applied.

In the case of no le or ge command, the prefix length must match exactly the length specified in the prefix list.

ip prefix-list description

`ip prefix-list NAME description DESC`

Descriptions may be added to prefix lists. This command adds a description to the prefix list.

Showing ip prefix-list

`show ip prefix-list [json]`

Display all IP prefix lists.

If the `json` option is specified, output is displayed in JSON format.

`show ip prefix-list NAME [json]`

Show IP prefix list can be used with a prefix list name.

If the `json` option is specified, output is displayed in JSON format.

`show ip prefix-list NAME seq NUM [json]`

Show IP prefix list can be used with a prefix list name and sequential number.

If the `json` option is specified, output is displayed in JSON format.

`show ip prefix-list NAME A.B.C.D/M`

If the command longer is used, all prefix lists with prefix lengths equal to or longer than the specified length will be displayed. If the command first match is used, the first prefix length match will be displayed.

`show ip prefix-list NAME A.B.C.D/M longer`

`show ip prefix-list NAME A.B.C.D/M first-match`

`show ip prefix-list summary [json]`

`show ip prefix-list summary NAME [json]`

`show ip prefix-list detail [json]`

`show ip prefix-list detail NAME [json]`

`debug prefix-list NAME match <A.B.C.D/M|X:X::X:X/M> [address-mode]`

Execute the prefix list matching code for the specified list and prefix. Shows which entry matched, if any. (`address-mode` is used for PIM RP lookups and skips prefix length checks.)

The return value from this command is success only if the prefix-list result is to permit the prefix, so the command can be used in scripting.

Clear counter of ip prefix-list

`clear ip prefix-list [NAME [A.B.C.D/M]]`

Clears the counters of all IP prefix lists. Clear IP Prefix List can be used with a specified NAME or NAME and prefix.

2.5 Route Maps

Route maps provide a means to both filter and/or apply actions to route, hence allowing policy to be applied to routes.

For a route reflector to apply a `route-map` to reflected routes, be sure to include `bgp route-reflector allow-outbound-policy` in `router bgp` mode.

Route maps are an ordered list of route map entries. Each entry may specify up to four distinct sets of clauses:

Matching Conditions A route-map entry may, optionally, specify one or more conditions which must be matched if the entry is to be considered further, as governed by the Match Policy. If a route-map entry does not explicitly specify any matching conditions, then it always matches.

Set Actions A route-map entry may, optionally, specify one or more Set Actions to set or modify attributes of the route.

Matching Policy This specifies the policy implied if the *Matching Conditions* are met or not met, and which actions of the route-map are to be taken, if any. The two possibilities are:

- *permit*: If the entry matches, then carry out the *Set Actions*. Then finish processing the route-map, permitting the route, unless an *Exit Policy* action indicates otherwise.
- *deny*: If the entry matches, then finish processing the route-map and deny the route (return *deny*).

The *Matching Policy* is specified as part of the command which defines the ordered entry in the route-map. See below.

Call Action Call to another route-map, after any *Set Actions* have been carried out. If the route-map called returns *deny* then processing of the route-map finishes and the route is denied, regardless of the *Matching Policy* or the *Exit Policy*. If the called route-map returns *permit*, then *Matching Policy* and *Exit Policy* govern further behaviour, as normal.

Exit Policy An entry may, optionally, specify an alternative *Exit Policy* to take if the entry matched, rather than the normal policy of exiting the route-map and permitting the route. The two possibilities are:

- *next*: Continue on with processing of the route-map entries.
- *goto N*: Jump ahead to the first route-map entry whose order in the route-map is >= N. Jumping to a previous entry is not permitted.

The default action of a route-map, if no entries match, is to deny. I.e. a route-map essentially has as its last entry an empty *deny* entry, which matches all routes. To change this behaviour, one must specify an empty *permit* entry as the last entry in the route-map.

To summarise the above:

	Match	No Match
Permit	action	cont
Deny	deny	cont

action

- Apply *set* statements
- If *call* is present, call given route-map. If that returns a `deny`, finish processing and return `deny`.
- If *Exit Policy* is *next*, goto next route-map entry
- If *Exit Policy* is *goto*, goto first entry whose order in the list is >= the given order.
- Finish processing the route-map and permit the route.

deny The route is denied by the route-map (return `deny`).

cont goto next route-map entry

`show route-map [WORD] [json]`

Display data about each daemons knowledge of individual route-maps. If WORD is supplied narrow choice to that particular route-map.

If the `json` option is specified, output is displayed in JSON format.

`clear route-map counter [WORD]`

Clear counters that are being stored about the route-map utilization so that subsuquent show commands will indicate since the last clear. If WORD is specified clear just that particular route-map's counters.

2.5.1 Route Map Command

`route-map ROUTE-MAP-NAME (permit|deny) ORDER`

Configure the *order*'th entry in *route-map-name* with `Match Policy` of either *permit* or *deny*.

2.5.2 Route Map Match Command

`match ip address ACCESS_LIST`

Matches the specified *access_list*

`match ip address prefix-list PREFIX_LIST`

Matches the specified *PREFIX_LIST*

`match ip address prefix-len 0-32`

Matches the specified *prefix-len*. This is a Zebra specific command.

`match ipv6 address ACCESS_LIST`

Matches the specified *access_list*

`match ipv6 address prefix-list PREFIX_LIST`

Matches the specified *PREFIX_LIST*

`match ipv6 address prefix-len 0-128`

Matches the specified *prefix-len*. This is a Zebra specific command.

`match ip next-hop address IPV4_ADDR`

This is a BGP specific match command. Matches the specified *ipv4_addr*.

`match ipv6 next-hop IPV6_ADDR`

This is a BGP specific match command. Matches the specified *ipv6_addr*.

`match as-path AS_PATH`
Matches the specified *as_path*.

`match metric METRIC`
Matches the specified *metric*.

`match tag TAG`
Matches the specified tag value associated with the route. This tag value can be in the range of (1-4294967295).

`match local-preference METRIC`
Matches the specified *local-preference*.

`match community COMMUNITY_LIST`
Matches the specified *community_list*

`match peer IPV4_ADDR`
This is a BGP specific match command. Matches the peer ip address if the neighbor was specified in this manner.

`match peer IPV6_ADDR`
This is a BGP specific match command. Matches the peer ipv6 address if the neighbor was specified in this manner.

`match peer INTERFACE_NAME`
This is a BGP specific match command. Matches the peer interface name specified if the neighbor was specified in this manner.

`match source-protocol PROTOCOL_NAME`
This is a ZEBRA specific match command. Matches the originating protocol specified.

`match source-instance NUMBER`
This is a ZEBRA specific match command. The number is a range from (0-255). Matches the originating protocols instance specified.

2.5.3 Route Map Set Command

`set tag TAG`
Set a tag on the matched route. This tag value can be from (1-4294967295). Additionally if you have compiled with the `--enable-realms` configure option. Tag values from (1-255) are sent to the Linux kernel as a realm value. Then route policy can be applied. See the tc man page.

`set ip next-hop IPV4_ADDRESS`
Set the BGP nexthop address to the specified IPV4_ADDRESS. For both incoming and outgoing route-maps.

`set ip next-hop peer-address`
Set the BGP nexthop address to the address of the peer. For an incoming route-map this means the ip address of our peer is used. For an outgoing route-map this means the ip address of our self is used to establish the peering with our neighbor.

`set ip next-hop unchanged`
Set the route-map as unchanged. Pass the route-map through without changing it's value.

`set ipv6 next-hop peer-address`
Set the BGP nexthop address to the address of the peer. For an incoming route-map this means the ipv6 address of our peer is used. For an outgoing route-map this means the ip address of our self is used to establish the peering with our neighbor.

`set ipv6 next-hop prefer-global`
For Incoming and Import Route-maps if we receive a v6 global and v6 LL address for the route, then prefer to use the global address as the nexthop.

set ipv6 next-hop global IPV6_ADDRESS
Set the next-hop to the specified IPV6_ADDRESS for both incoming and outgoing route-maps.

set local-preference LOCAL_PREF
Set the BGP local preference to *local_pref*.

set local-preference +LOCAL_PREF
Add the BGP local preference to an existing *local_pref*.

set local-preference -LOCAL_PREF
Subtract the BGP local preference from an existing *local_pref*.

set distance DISTANCE
Set the Administrative distance to DISTANCE to use for the route. This is only locally significant and will not be dispersed to peers.

set weight WEIGHT
Set the route's weight.

set metric <[+|-](1-4294967295)|rtt|+rtt|-rtt>
Set the BGP attribute MED to a specific value. Use +/- to add or subtract the specified value to/from the MED. Use *rtt* to set the MED to the round trip time or *+rtt*/*-rtt* to add/subtract the round trip time to/from the MED.

set as-path prepend AS_PATH
Set the BGP AS path to prepend.

set as-path exclude AS-NUMBER...
Drop AS-NUMBER from the BGP AS path.

set community COMMUNITY
Set the BGP community attribute.

set ipv6 next-hop local IPV6_ADDRESS
Set the BGP-4+ link local IPv6 nexthop address.

set origin ORIGIN <egp|igp|incomplete>
Set BGP route origin.

set table (1-4294967295)
Set the BGP table to a given table identifier

set sr-te color (1-4294967295)
Set the color of a SR-TE Policy to be applied to a learned route. The SR-TE Policy is uniquely determined by the color and the BGP nexthop.

2.5.4 Route Map Call Command

call NAME
Call route-map *name*. If it returns deny, deny the route and finish processing the route-map.

2.5.5 Route Map Exit Action Command

`on-match next`

`continue`

Proceed on to the next entry in the route-map.

`on-match goto N`

`continue N`

Proceed processing the route-map at the first entry whose order is >= N

2.5.6 Route Map Optimization Command

`route-map ROUTE-MAP-NAME optimization`

Enable route-map processing optimization for *route-map-name*. The optimization is enabled by default. Instead of sequentially passing through all the route-map indexes until a match is found, the search for the best-match index will be based on a look-up in a prefix-tree. A per-route-map prefix-tree will be constructed for this purpose. The prefix-tree will compose of all the prefixes in all the prefix-lists that are included in the match rule of all the sequences of a route-map.

2.5.7 Route Map Examples

A simple example of a route-map:

```
route-map test permit 10
 match ip address 10
 set local-preference 200
```

This means that if a route matches ip access-list number 10 it's local-preference value is set to 200.

See *Miscellaneous Configuration Examples* for examples of more sophisticated usage of route-maps, including of the `call` action.

2.6 IPv6 Support

FRR fully supports IPv6 routing. As described so far, FRR supports RIPng, OSPFv3, and BGP-4+. You can give IPv6 addresses to an interface and configure static IPv6 routing information. FRR IPv6 also provides automatic address configuration via a feature called `address auto configuration`. To do it, the router must send router advertisement messages to the all nodes that exist on the network.

Previous versions of FRR could be built without IPv6 support. This is no longer possible.

2.6.1 Router Advertisement

`show ipv6 nd ra-interfaces [vrf <VRFNAME|all>]`

Show configured route advertisement interfaces. VRF subcommand only applicable for netns-based vrfs.

`ipv6 nd suppress-ra`

Don't send router advertisement messages. The `no` form of this command enables sending RA messages.

`ipv6 nd prefix ipv6prefix [valid-lifetime] [preferred-lifetime] [off-link] [no-autoconfig] [router-addre`

Configuring the IPv6 prefix to include in router advertisements. Several prefix specific optional parameters and flags may follow:

- `valid-lifetime`: the length of time in seconds during what the prefix is valid for the purpose of on-link determination. Value `infinite` represents infinity (i.e. a value of all one bits (`0xffffffff`)). Range: `(0-4294967295)` Default: `2592000`
- `preferred-lifetime`: the length of time in seconds during what addresses generated from the prefix remain preferred. Value `infinite` represents infinity. Range: `(0-4294967295)` Default: `604800`
- `off-link`: indicates that advertisement makes no statement about on-link or off-link properties of the prefix. Default: not set, i.e. this prefix can be used for on-link determination.
- `no-autoconfig`: indicates to hosts on the local link that the specified prefix cannot be used for IPv6 autoconfiguration.

  Default: not set, i.e. prefix can be used for autoconfiguration.
- `router-address`: indicates to hosts on the local link that the specified prefix contains a complete IP address by setting R flag.

  Default: not set, i.e. hosts do not assume a complete IP address is placed.

`ipv6 nd ra-interval [(1-1800)]`

The maximum time allowed between sending unsolicited multicast router advertisements from the interface, in seconds. Default: `600`

`ipv6 nd ra-interval [msec (70-1800000)]`

The maximum time allowed between sending unsolicited multicast router advertisements from the interface, in milliseconds. Default: `600000`

`ipv6 nd ra-fast-retrans`

RFC4861 states that consecutive RA packets should be sent no more frequently than three seconds apart. FRR by default allows faster transmissions of RA packets in order to speed convergence and neighbor establishment, particularly for unnumbered peering. By turning off ipv6 nd ra-fast-retrans, the implementation is compliant with the RFC at the cost of slower convergence and neighbor establishment. Default: enabled

`ipv6 nd ra-retrans-interval [(0-4294967295)]`

The value to be placed in the retrans timer field of router advertisements sent from the interface, in msec. Indicates the interval between router advertisement retransmissions. Setting the value to zero indicates that the value is unspecified by this router. Must be between zero or 4294967295 msec. Default: `0`

`ipv6 nd ra-hop-limit [(0-255)]`

The value to be placed in the hop count field of router advertisements sent from the interface, in hops. Indicates the maximum diameter of the network. Setting the value to zero indicates that the value is unspecified by this router. Must be between zero or 255 hops. Default: `64`

`ipv6 nd ra-lifetime [(0-9000)]`

The value to be placed in the Router Lifetime field of router advertisements sent from the interface, in seconds. Indicates the usefulness of the router as a default router on this interface. Setting the value to zero indicates that the router should not be considered a default router on this interface. Must be either zero or between value specified with `ipv6 nd ra-interval` (or default) and 9000 seconds. Default: `1800`