

Chapter 7

GateD Interactive Interface (GII)

7.1 Overview

The GateD Interactive Interface (GII) provides an interactive interface to a running GateD daemon. This interface can be used to query internal GateD variables. GII, which is implemented like any other protocol in GateD, accepts telnet connections to port 616 (C.f. [RFC 854] for a description of the telnet protocol). After user identification, GII answers any query sent as ASCII commands. The commands include queries about the memory, routing table, interface list, and other internal parameters.

7.2 Configuration

GII will identify the user by using the UNIX password of the GII ID of the system. In other words, when using the GII interface, one must create a GII account by editing the `/etc/passwd` file.

Alternatively, any user name can be used by changing `GII_USER` in the `gii.h` file.

7.3 Connecting to GII

1. Open a telnet connection to the machine running GateD on TCP port 616.
2. Identify yourself using a simple password.
3. Enter `quit` to end your telnet session.

Here is an example of such a session:

```
% telnet router1.foo.bar 616
Trying 192.168.10.1...
Connected to router1.foo.bar.
Password?
100 GateD Interactive Interface. Version 4-0-6
GateD-router1.foo.bar> help
100 HELP: The possible commands are:
100     help: Print help messages
100     show: Show internal values
100     quit: Close the session
GateD-router1.foo.bar> show
100 HELP: The possible subcommands are:
```

```
100    version: Show the current GateD version
100    kernel: Show the Kernel support
100    interface [name|index]: Show interface status
100    memory: Show the memory allocation
100    ip: Show info about IP protocol
100    task: Show list of active tasks
100    dvmrp: Show info about DVMRP protocol
100    ospf: Show info about OSPF protocol
100    timer: Show list of timers
100    bgp: Show info about BGP protocol
100    rip: Show info about RIP protocol
100    ripng: Show info about RIPng protocol
GateD-router1.foo.bar> quit
```

7.4 Show Command Parameters

show version

show version displays the current GateD version.

show kernel

show kernel displays the kernel support.

show interface [name | index]

show interface displays the interface status on the interface specified.

show memory

show memory displays the allocation of memory blocks used by GateD.

show ip [route | walkup | walkdown | rpf] prefix/len

show ip displays the IPv4 routes in the GateD routing table.

show task

show task displays a list of active tasks.

show ip mroute [boundaries | mfc | static]

show ip mroute displays multicast routing information based on administrative boundaries, multicast forwarding cache or static routes.

show ospf [global | interface address | area area-ID [LS_type]]

Only OSPF version 2 is supported.

show timer

timer displays a list of timers.

show ip igmp groups

show ip igmp groups displays directly connected groups learned via IGMP.

show bgp [aspath [regexp] | cidr-only [prefix/masklen] | community community number | peers as_no | peer-group [internal | external | internal_igp | routing | test] | routes prefix/masklen | summary]

show bgp displays information about BGP routes. See page 24, "BGP", for details on the BGP commands.

```
show rip [ routes [ prefix/masklen ] | summary | tag [ tag ] ]
```

`show rip` displays information about RIP routes.

7.5 GateD Core

The following commands will display parameters related to the internal core of GateD:

```
show version
```

`show version` displays the version of the running GateD.

```
show kernel
```

`show kernel` displays the type of kernel of the host, including which features are supported (for example, reject routes, Multicast, UDP checksums, and so on).

```
show memory
```

`show memory` displays the memory usage, divided by memory block structures.

```
show task
```

`show task` displays the running tasks in GateD.

```
show timers
```

`show timers` displays a list of timers.

7.6 Interfaces

The following command will display parameters related to the interfaces:

```
show interface [ name | index ]
```

Without a parameter, this command lists all the interfaces of the system. If one interface name or index is given as an argument, all the parameters concerning this interface are displayed.

7.7 Routing tables

The following commands will display parameters related to the routing table:

```
show ip route [ prefix/len ]
```

Without a parameter, `show ip route` prints the size of the IP routing table. With a parameter, which must be a prefix number and mask length, `show ip route` displays complete information about the given route, such as the number of announcements, next hop, AS path, active route, and so on.

```
show ip walkup [ prefix/len ]
```

`show ip walkup` lists all the routes that are less specific than *prefix/len*, for example, all the components of the aggregate *prefix/len*.

```
show ip walkdown [ prefix/len ]
```

`show ip walkdown` lists all the routes that are more specific than *prefix/len*, for example, all the components of the aggregate *prefix/len*. `show ip walkdown 0/0` will display the whole routing table. `^C` can be used to stop the listing.

```
show ip6 walkup [ prefix/len ]
```

`show ip6 walkup` lists all the routes that are less specific than *prefix/len*. For IPv6, for example, all the components of the aggregate *prefix/len*.

show ip6 walkdown [*prefix/len*]

show ip6 walkdown lists all the routes that are more specific than *prefix/len*. For IPv6, for example, all the components of the aggregate *prefix/len*. **show ip walkdown 0/0** will display the whole routing table. ^c can be used to stop the listing.

show ip rpf [*prefix/len*]

show ip rpf [*prefix/len*] displays the RPF information.

7.8 RIP

show rip routes [*prefix/masklen*]

show rip routes displays RIP information about routes.

show rip summary

show rip summary displays a table of information about all RIP routes.

show rip tag [*tag*]

show rip tag displays RIP tag information.

7.9 OSPF

show ospf global

show ospf global displays general OSPF options.

show ospf interface address

show ospf interface address displays IP interface status.

show ospf area *area-ID* [*LS_TYPE*]

show ospf area *area-ID* displays area information.

show ospf asex

show ospf asex displays the contents of the router-global ASEX (type-5) LSA Database.

show ospf asopaque

show ospf asopaque displays the contents of the router-global AS-Scope Opaque LSA (type-11) Database.

7.10 ISIS

show isis global

show isis global displays some global ISIS parameters

show isis interface *address*

show isis interface *address* displays the interface status.

7.11 BGP

show bgp aspath [*regex*]

show bgp aspath displays **bgp aspath**'s that match the regular expression. If null, all paths should be sent.

show bgp cidr-only [*prefix/masklen*]

show bgp cidr-only prints only CIDR routes based the route table. Class prefixes of /8 , /16, and /24 are ignored. If no routes are specified, the tree walks from the top of the route table. If not, it walks from the **prefix/masklen**.

show bgp community [*community number*]
show bgp community displays BGP routes associated with the community number specified. The community number is specified as a hexadecimal number.

show bgp peeras *as_num*
show bgp peeras *as_num* displays BGP peer information associated with this AS, such as **bgp peer address**, **bgp versions**, **gateway** (third party). *as_num* is the AS number of the peers you want to track. An *as_num* must be specified or an error will result.

show bgp peer-group [**internal** | **external** | **internal_igp** | **routing** | **test**]
show bgp peer-group displays summary information about all BGP peers in grouping, such as:
peer info - AS, BGP version, neighbor addresses
statistics - number of updates in/out, state

show bgp routes *prefix/masklen*
show bgp routes displays BGP information about these routes.

show bgp summary
show bgp summary displays summary information about all BGP peers, such as:
peer info - AS, BGP version, neighbor addresses
statistics - number of updates in/out, state

7.12 IGMP

show ip igmp groups displays directly connected groups learned via IGMP.

7.13 DVMRP

show ip dvmrp interfaces shows interfaces configured for DVMRP.

show ip dvmrp mfc shows DVMRP forwarding entries.

show ip dvmrp neighbors shows DVMRP neighbor information.

show ip dvmrp errors shows DVMRP RPF lookup errors.

show ip dvmrp route *ip_address* shows the DVMRP route for *ip_address*.

show ip dvmrp routes [*network*][/*network-mask*] shows all routes learned via DVMRP or the route given by [*network*][/*network-mask*].

show ip dvmrp targets shows information about DVMRP targets.

show ip dvmrp df[*network/netmask*] shows the DVMRP designated forwarder for [*network/netmask*] on all downstream interfaces.

show ip dvmrp prunes shows DVMRP prunes received.

show ip dvmrp prunexmits shows DVMRP prunes sent.

7.14 PIM-SM

show ip pim bsr displays BSR status.

show ip pim crp displays the crp status.

show ip pim interface displays pim interface information.

show ip pim neighbor displays PIM-SM neighbor information.

show ip pim route displays PIM-SM routing tables.

show ip pim rp-hash *ip-address* displays the group hash mapping for *ip-address*.

`show ip pim rpset` displays the RendezvousPoint-Set.

`show ip pim timeouts` displays all component timeouts.

`show ip pim walkup prefix/masklen` displays less specific routes for multicast.

`show ip pim walkdown prefix/masklen` displays more specific routes for multicast.

7.15 Multicast Routing Table

`show ip mroute boundaries`
`show ip mroute boundaries` displays administratively-scoped boundary information.

`show ip mroute mfc`
`show ip mroute mfc` displays forwarding cache information.

`show ip mroute static`
`show ip mroute static` displays static group membership information.

7.16 Miscellaneous GII Commands

`gii-clear-interface-times machine` clears the interface timers that record the first and last interface change.

`gii-diff-interface-times machine` shows the time elapsed between first learned and last learned interfaces

`gii-diff-routes-learned-time machine first_route/masklen last_route/masklen` shows the time elapsed between two learned routes. The routes should be in the format route/masklength.

`gii-diff-routes-update-time machine first_route/masklen last_route/masklen` shows the time elapsed between two routes that were updated. The routes should be in the format route/masklength.

`gii-get-route-learned-time machine [route]` displays the time a route was learned. If no route was specified, a list of all routes in GateD will be displayed.

`gii-show-interface-times machine` shows interface times. The first interface time is the first learned interface, and the second interface time is the last learned interface.

7.17 Interface Commands

`add-multiple-aliases [amount]` adds simultaneously to each machine specified by the set-machines command aliases with the subnet of 10.128/9. Rack number and machine letter is used to obtain unique addresses.

`add-multiple-aliases machine if_name start_addr num_aliases [offset]` adds a sequence of aliases on machine *machine* and associates the addresses with a specific interface *if_name*. The *start_addr* represents the address. If no mask was specified, a mask of length 24 is used as the default value.

If this command is used in tunnel sequence settings, the keyword **other** can be used to specify the other side of the tunnel. In this case the addresses are offset by 0.1.0.0 from the specified *start_addr*.

`add-multiple-tunnels machine if_name start_addr num_tunnels [option] [offset]` adds a sequence of tunnels on machine *machine* and associates the addresses with a specific interface *if_name*. The *start_addr* represents the address of the alias to associate with the tunnel. All tunnel addresses are offset from this address by 0.10.0.0. This command associates

the tunnel with the alias address, unless otherwise specified by the **single** keyword, which will then read the interface IP address and associate the tunnel to it.

To set the other side of the tunnel, the **other** keyword is used without changing the **start_addr** of the first side.

If x gif tunnels already exist, you can specify the offset number to leave the first **[offset]** gifs intact.

add-multiple-tunnels-verbose *machine if_name start_addr num_tunnels [option]* **[offset]** is equivalent to **add-multiple-tunnels** except that it displays additional information about time and memory values.

flap-tunnels *interface starting_address amount count [offset]* will flap tunnels by adding *amount* gif interfaces and subsequently removing the interfaces *count* number of times. **[offset]** again specifies here whether the tunnels will be created from a given gif address. These tunnels cannot be used for routing since they set only one side of the machine and use the IP address of *interface* to associate the tunnels to.

updown-tunnels *machine interface starting_address amount count [offset]* is similar to the **flap-tunnels** command with the difference that tunnels are not created, but the gif interfaces are set up and down.

remove-multiple-aliases *machine if_name start_addr num_tunnels [option] [offset]* will remove *num_tunnels* aliases sequentially with addresses starting at *start_addr* on interface *if_name*.

remove-multiple-tunnels *machine if_name start_addr num_tunnels [offset]* will remove *num_tunnels* tunnels sequentially starting at **[offset]**, which represents the gif number. If no offset is specified the default value '0' is used. Although the start address and interface are required by the command, it isn't really used and is left here for consistent syntax with the rest of the tunnel commands.

7.18 Status Commands

compare-tunnels **[machines]** is similar to standard compare function, but differs in that in addition to comparing routes, it also compares tunnels. It does not have the capability to compare ribs.

compare **[machines]** runs compare results on **[machines]**.

