Chapter 36
Frequently Asked Questions

36.1 Kernel Interactions

1. Does GateD support dynamic configuration on the fly? Does GateD support command line coding?

GateD uses a flat file, rather than a command line interface. To re-read the file after configuration changes, GateD must receive a SIGHUP. This can be accomplished either directly through the UNIX signaling mechanism or by use of the GDC (GateD Controller). However, this configuration is "dynamic," in that states are retained between reconfigs whenever possible. For example, BGP connections are not terminated.

2. Does GateD use the UNIX mbuf structure or its own buffer?

GateD does not use UNIX mbuf structures. GateD uses its own buffers, dynamically allocated from process address space.

3. Does GateD require pre-emptive (such as UNIX, Windows, or Linux) or non-preemptive operating systems (such as PSOS and any real-time embedded systems)?

A complete listing of currently supported operating systems is available. GateD runs in a single process with co-operative multitasking inside the processes. We do not require preemptive or non-preemptive operating systems as long as the process receives adequate cycles.

4. Does GateD require a UNIX-style "fork" function?

The "**`dump`**" feature in GateD uses **`fork(2)`** to spawn a child process. The child dumps internal state to a file and exits. If this feature is not necessary, it may be removed without affecting operation.

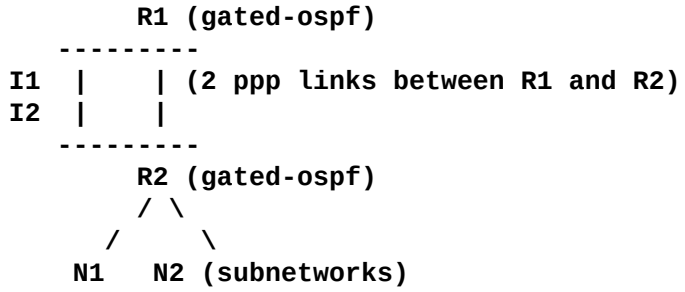
5. Does GateD require timer functions from the operating system (for example, 1/10 or 1/100 sec time granularity, recursive timer, sleep, signal, and so on)?

In releases that support signal I/O and interval timer, the **SIGALARM** and **SIGIO** signals may be sent by the operating system to allow GateD to respond to certain critical events, for example, expiration of an OSPF **HELLO** timer. With normal timers, a timer granularity of 1/10 second is sufficient for operation.

36.2 Protocols

36.2.1 OSPF

1. In a network with the following topology:



there are two point-to-point links, I1 and I2, between routers R1 and R2. R2 is connected to networks N1 and N2. Both routers are running gated-ospf. How can I route the traffic from R1 to R2 on I1 only if it is destined to N1, and on I2 only if it is destined to N2?

Put I2 and N2 in one area and I1 and N1 in another area. OSPF prefers intra-area routes over inter-area. An example configuration would be:

For R1:

```
ospf yes {
    priority 1;
    backbone {
        interface I1;
    };
    area 0.0.0.1 {
        interface I2;
    };
};
```

For R2:

```
ospf yes {
```

```
        priority 1;
        backbone {
            interface I1;
            interface N1;
        };
        area 0.0.0.1 {
            interface I2;
            interface N2;
        };
    };
```

2. The following message appears when GateD is restarted. "gated[28100]: task_get_proto: getprotobyname("ospf") failed, using proto 89" What does this message mean?

The `/etc/protocol` file doesn't contain an ospf entry. The entry should look something like this:

```
ospf    89  OSPFIGP     # Open Shortest Path First IGP
```

The 89 is the assigned Internet protocol number specified in RFC 1700. GateD will use the default value and continue if possible.

3. OSPF does not seem to install multiple equal-cost routes to a destination. I know OSPF supports ECMP (Equal-Cost Multipath Routing). What's wrong?

The OSPF module supports ECMP by installing more than one next hop in the RIB. Kernel support for this feature, however, is not widely available. In order for all of the equal-cost routes to a destination to be installed, the krt_rt_xxx module (and kernel) must support this.

36.2.2 BGP

1. How can I increase the number of BGP peers GateD will allow?

Specify a value for **`RTBIT_SIZE`** in your config file and recompile GateD. Each increment of **`RTBIT_SIZE`** provides 32 additional bits for 32 additional peers.

Example: **`options RTBIT_SIZE=4`** will allow up to 128 peers. The default value of **`RTBIT_SIZE`** is 1.

2. Why is GateD changing the NEXT_HOP attribute when advertising a route to an internal peer?

The BGP RFC states: “When a BGP speaker advertises the route to another BGP speaker located in its own autonomous system, the advertising speaker shall not modify the NEXT_HOP attribute associated with the route.” See “Chapter 14 Border Gateway Protocol (BGP)” on page 61 for more information about the BGP statement.

Basically, GateD is designed not to modify the NEXT_HOP if it believes that its IBGP peer will be able to figure out how to reach the address it depicts. GateD will go ahead and rewrite the NEXT_HOP if it believes that the peer will not know how to reach the depicted address.

In the case of group type internal, GateD knows (because you have so configured it) that its peers do not do BGP-IGP next hop resolution. GateD also knows that all of its peers are L2-adjacent, so it rewrites the NEXT_HOP to something it knows its peer will be able to reach at L2.

If you wish to make GateD conform to the RFC instead of allowing this behavior, you can use group type routing with **`interface all`** specified. In the case of **`group type routing ... interface all`**, GateD knows that (a) its peers are resolving BGP-IGP (this is a property of group type routing) and (b) NEXT_HOPs via any interface are known via the IGP (this is what "**`interface all`**" means). So it won't rewrite any NEXT_HOPs.

3. I keep seeing error messages about an unsupported optional parameter when trying to peer with a Cisco®. What is the problem?

Some versions of Cisco® IOS have a capabilities negotiation bug. The intended behavior of capabilities negotiation is to resend a BGP open message without the optional parameter once it receives a notification from GateD stating that the parameter is unsupported. You should upgrade your Cisco® or apply this workaround: **`neighbor x.x.x.x dont-capability-negotiate`**. For more information about capabilities negotiation, refer to **`draft-ietf-idr-bgp4-cap-neg-03.txt`**.

4. How can I configure a peer that is not on the same network?

Use the **`gateway`** keyword on the peer statement:

```
group type external peeras 65000 {
    peer a.b.c.d gateway w.x.y.z;
};
```

where *`a.b.c.d`* is your peer's IP address and *`w.x.y.z`* is the next hop that GateD should use to find *`a.b.c.d`*.

5. Why isn't BGP advertising my static routes?

If no export policy is specified, BGP will advertise only direct (interface) routes. To export static routes, you will need an export statement like this:

```
export proto bgp as 65500 {
    proto static {
```

```
            all;
        };
    };
```

Note that once export policy has been defined for BGP, GateD needs to be explicitly configured in order to export direct (interface) routes. Use proto direct to do this. See “Exporting to BGP” on page 149 for more information on configuring BGP export policy.

6. Why is GateD ignoring my MEDs?

If you want GateD to pay attention to incoming metrics, you need to specify the **med** keyword on the group statement:

```
group type external peeras 65530 med {
    peer 192.168.10.2;
};
```

The default behavior is for GateD to ignore incoming metrics.

7. What is BGP's default import and export behavior?

GateD will import all routes from a configured peer unless otherwise configured. If no export policy is specified, BGP will advertise only direct (interface) routes. See “Examples of Importation into Multicast RIBs” on page 142 and “Exporting to BGP” on page 149 for more information on configuring BGP policy.

8. If, on a Cisco® router, a route is redistributed (exported) from another protocol, such as static or OSPF, into BGP, what should the origin of the route be?

If you redistribute using the "network x.x.x.x" command, your Cisco® router will automatically set the origin to "IGP." If you use "redistribute," your Cisco® router will use origin incomplete. Either will work. A description of the origin path attribute follows.

ORIGIN (Type Code 1):

ORIGIN is a well-known mandatory attribute that defines the origin of the path information. The data octet can assume the following values:

Value	Meaning
0	IGP - Network Layer Reachability Information is interior to the originating AS.
1	EGP - Network Layer Reachability Information is learned via EGP.

Value	Meaning
2	INCOMPLETE - Network Layer Reachability Information is learned by some other means.