

# Bridge design issues

---

*Source:* <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/net/2008-02/msg00325.html>

---

- *From:* Barney Cordoba <[barney\\_cordoba@xxxxxxxxx](mailto:barney_cordoba@xxxxxxxxx)>
  - *Date:* Fri, 29 Feb 2008 06:28:11 -0800 (PST)
- 

I've been playing with the bridging code, and something doesn't seem quite right about it. It seems to work better and more efficiently when configured the "wrong" way, according to the handbook.

The "correct" way would be a setup as follows:

```
bridge0 1.2.3.4 netmask 255.255.255.0 addm em0 addm em1 up
```

where the bridge has the address and em0 and em1 are members of the bridge. However this results in some ugliness:

The path for traffic originating from the system itself goes through the standard ethernet output machinery and gets queued twice:

```
ip_output(bridge0) -> ether_output(bridge0) -> queued on bridge0 :: bridge_start() ->bridge_enqueue(em0/em1)
```

bridge\_output() is never used in this setup:

When the following "wrong" setup is used:

```
ifconfig em0 1.2.3.4 netmask 255.255.255.0  
ifconfig bridge0 addm em0 addm em1 up
```

Where em0 has the address. This results in 2 desirable things: 1) a real ethernet is used in packets, and 2) the path is streamlined:

```
ip_output(bridge0) -> ether_output(bridge0) ->  
bridge_output(bridge0) -> bridge_enqueue(em0/em1)
```

In my view, it should work the same no matter what the setup, if done correctly. Any output sent to a bridge should be passed through the same bridge output

## Bridge design issues

machinery. So

ether\_output(em0|em1|bridge0) should all resolve to bridge\_output(bridge0) and let the bridge determine the proper path.

The reason the FreeBSD code doesn't work this way is that the bridge interface doesn't point to itself.

Setting if\_bridge in the bridge interface structure to point to itself seems to normalized the path.

bridge\_output() needs to be made aware that it may be passed a bridge interface to make sure it resolves the route before trying to send the packet.

bridge\_forward() should also use bridge\_output() rather than resolving itself. The bridge resolution is always the same logic; its undesirable to have different paths for every case. Arguably, packets that arrive from the local stack are no different from ones that arrive via a bridge member in terms of output resolution.

There should also be a way to assign a static ethernet address to a bridge interface so its always the same.

ISPs in many countries are required to keep track of devices so random ethernets are problematic idea for them

Barney

---

Looking for last minute shopping deals?

Find them fast with Yahoo! Search.

<http://tools.search.yahoo.com/newsearch/category.php?category=shopping>

---

freebsd-net@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-net>

To unsubscribe, send any mail to "freebsd-net-unsubscribe@xxxxxxxxxxx"