

## ng\_one2many v.s. AFT (NIC Fault Tolerance/Fail Over/Redundancy Revisited)

**Source:** <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/questions/2005-10/1354.html>

---

**From:** Brian A. Seklecki ([lavalamp\\_at\\_spiritual-machines.org](mailto:lavalamp_at_spiritual-machines.org))

**Date:** 10/16/05

Date: Sat, 15 Oct 2005 19:25:14 -0400 (EDT)

To: Jonathan Donaldson <[donaldson@cisco.com](mailto:donaldson@cisco.com)>, Brad Bendy <[brad@shockwebhost.com](mailto:brad@shockwebhost.com)>, [jks@clickcom.co](mailto:jks@clickcom.co)

Re:

<http://lists.freebsd.org/pipermail/freebsd-questions/2005-October/100623.html>

First: This is all very preliminary from some testing over the weekend.

Dell's reponse was that Intel's AFT/ALB was entirely software based.

That left me with few options:

- 1) Try userland layer 3 failover (ugly)
- 2) Use ng\_one2many

However, ng\_one2many only permits for two algorithms:

NG\_ONE2MANY\_XMIT\_ROUNDROBIN and NG\_ONE2MANY\_XMIT\_ALL.

However, none of these meet the need:

- Round-Robin results in 50% packet loss if a hook/interface is lost (not acceptable in any mission critical environment).
- Xmit-All causes twice as much load on to be placed on the switch /fabric and switch CPU.

What ng\_one2many needs is a "Active-Standy" XMIT algorithm (STP BOFH's will think BLOCKING/FORWARDING). It could even be used on top of other NetGraph nodes like ng\_fec or possibly (hopefully) ng\_802.3ad >:}

Essentially, a single layer 3 IP address needs to be visible in a "switch fault tolerant" or "adapter fault tolerant" configuration. A userland-level daemon could be scripted, and it has been done before:

<http://lists.freebsd.org/pipermail/freebsd-isp/2003-November/001314.html>

So when a fail-over occurs, the layer IP 3 address moves from one layer 2 MAC address to another layer 2 MAC address on the same machine (and same subnet, same ethernet segment, just a different interface). TCP sockets should not be affected due to layer abstraction.

## freebsd-questions: ng\_one2many v.s. AFT (NIC Fault Tolerance/Fail Over/Redundancy Revisited)

This got me thinking about HSRP/VRRP. That protocol is designed strictly to move a layer 3 address between two different hosts. Excellent applications are Router/Firewall and VPN concentrator, as OpenBSD's carp(4) has implemented with the help of pfsync. I was experimenting with the OpenBSD variant and I realized that client hosts weren't seeing the usual warnings about MAC address changes.

As of 3.7, OpenBSD's CARP shares a virtual MAC address between the hosts, Cisco's HSRP does not.

Then I was thinking about the OpenBSD/NetBSD bridge(4) interface. If the host acting as the bridge wishes too, it can participate in the bridged networks by assigning a layer 3 address. The address isn't ifconfig(8)'d do the "bridge0" interface. Instead, it's assigned to the first interface included in the "bridge[0-9]", say fxp0.

Further more, regardless of what network segment/port a host participating in a bridge(4)'d network resides, the ARP'd IP address of the OpenBSD/NetBSD host is persistently the MAC first physical interface ifconfig(8)'d with the IP.

Plus OpenBSD/NetBSD bridge(4) supports 802.1d spanning tree >:}

This is important. Spanning Tree as an alogirth could provide Intel AFT "Fault Tolerance" intelligence if the persistent layer2 address of a host was unchanged with the NIC interface change. The function of STP is to provide a loop free path to every layer2 MAC in a segment. But a STP enabled bridge(4) with an IP address assigned has a persistent MAC address associated with a layer 3 address!

Therefore, the solution has been there all along. The attached diagram explains in greater detail.

[http://digitalfreaks.org/~lavalamp/OpenBSD\\_Bridge\\_AFT.png](http://digitalfreaks.org/~lavalamp/OpenBSD_Bridge_AFT.png)

In this diagram, switch 0 is configured manually as the spanning tree root and switch 1 is the backup spanning tree root. By default, r10 will be in BLOCKING and r11 will being FORWARDING. However, as tcpdump(8) illustrates, regardless of which interface is the root port, ARP replys will always return the MAC if the bridge(4) member interface ifconfig(8)'d with the IP.

```
r10: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    address: 00:50:fc:9d:24:d6
    media: Ethernet autoselect (100baseTX full-duplex)
    status: active
    inet 192.168.100.1 netmask 0xfffff00 broadcast 192.168.100.255
```

```
r11: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    address: 00:50:fc:9d:08:cd
    media: Ethernet autoselect (100baseTX full-duplex)
```

## freebsd-questions: ng\_one2many v.s. AFT (NIC Fault Tolerance/Fail Over/Redundancy Revisited)

status: active

```
---
bridge0: flags=41<UP,RUNNING>
  Configuration:
    priority 32768 hellotime 2 fwddelay 15 maxage 20
  Interfaces:
    r11 flags=b<LEARNING,DISCOVER,STP>
      port 2 ifpriority 128 ifcost 55 forwarding
    r10 flags=b<LEARNING,DISCOVER,STP>
      port 1 ifpriority 128 ifcost 55 blocking
  Addresses (max cache: 100, timeout: 240):
    00:01:63:bb:f7:c9 r11 1 flags=0<>
    00:0f:1f:c1:f2:b7 r11 1 flags=0<>
-----
# tcpdump -i r11 -n arp
12:38:17.806885 arp who-has 192.168.100.1 tell 192.168.100.254
12:38:17.806951 arp reply 192.168.100.1 is-at 0:50:fc:9d:24:d6
12:38:17.806966 arp reply 192.168.100.1 is-at 0:50:fc:9d:24:d6
bs0#sh spanning-tree vlan 11 interface fa0/9
Spanning tree 11 is executing the IEEE compatible Spanning Tree protocol
  Bridge Identifier has priority 100, address 0001.63bb.f7c2
  Configured hello time 2, max age 20, forward delay 15
  We are the root of the spanning tree
  Topology change flag not set, detected flag not set, changes 54
  Times: hold 1, topology change 35, notification 2
    hello 2, max age 20, forward delay 15
  Timers: hello 0, topology change 0, notification 0
Interface Fa0/9 (port 22) in Spanning tree 11 is FORWARDING
  Port path cost 19, Port priority 128
  Designated root has priority 100, address 0001.63bb.f7c2
  Designated bridge has priority 100, address 0001.63bb.f7c2
  Designated port is 22, path cost 0
  Timers: message age 0, forward delay 0, hold 0
  BPDU: sent 10592, received 30
bs1#sh spanning-tree vlan 11 interface fa0/9
Spanning tree 11 is executing the IEEE compatible Spanning Tree protocol
  Bridge Identifier has priority 32768, address 0002.fd0e.f382
  Configured hello time 2, max age 20, forward delay 15
  Current root has priority 100, address 0001.63bb.f7c2
  Root port is 38, cost of root path is 19
  Topology change flag not set, detected flag not set, changes 54
  Times: hold 1, topology change 35, notification 2
    hello 2, max age 20, forward delay 15
  Timers: hello 0, topology change 0, notification 0
Interface Fa0/9 (port 22) in Spanning tree 11 is FORWARDING
  Port path cost 19, Port priority 128
  Designated root has priority 100, address 0001.63bb.f7c2
  Designated bridge has priority 32768, address 0002.fd0e.f382
  Designated port is 22, path cost 19
  Timers: message age 0, forward delay 0, hold 0
  BPDU: sent 45454, received 1196
bs0#sh mac-address-table | include 24d6
0050.fc9d.24d6      Dynamic          11  FastEthernet0/9
bs1#sh mac-address-table | include 24d6
0050.fc9d.24d6      Dynamic          11  FastEthernet0/24
The behavior is similar in FreeBSD using ng_bridge(4) (I haven't tried
FreeBSD bridge(4)). However, both of these claim "a privative loop
prevention algorithm"); ... 'debug stp events' shows no STP traffic from a
FreeBSD host, though.
Also, FreeBSD differs in behavior in that the MAC address ARP'd is that of
which ever NG node bridge member is assigned the IP.
```

## freebsd-questions: ng\_one2many v.s. AFT (NIC Fault Tolerance/Fail Over/Redundancy Revisited)

The disadvantage is that without FreeBSD speaking 802.1d, it can't know to fail an interface on any event other than a media state change. i.e., the currently active port could be connected a switch that loses its uplink. Of course, neither the FreeBSD or NetBSD/OpenBSD implementation features a "heartbeat" algorithm to add intelligence, as Intel AFT/ALB might, but that wasn't the design principal goal.

Also, my initial tests are with managed switches using PVST. Behavior may differ with unmanaged switches where no STP debugging is possible or possibly a uni-stp is used.

More on this on Monday...

[http://www.cisco.com/application/pdf/en/us/quest/netsol/ns304/c649/cdccont\\_0900aec800ea162.pdf~BAS](http://www.cisco.com/application/pdf/en/us/quest/netsol/ns304/c649/cdccont_0900aec800ea162.pdf~BAS)

---

freebsd-questions@freebsd.org mailing list

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

To unsubscribe, send any mail to "freebsd-questions-unsubscribe@freebsd.org"