# Solaris IP Multipathing made easy

I've recently setup a bunch of machines for IP multipathing (i.e. recent to this article - Nov 28, 2001) by following the Sun blueprint paper. I thought I would share a simpler step by step approach.

1. get 2 network interface cards in your machine (some machines, like Netra T1 series have 2 builtin). It is not required that they be the same type (e.g. Sun SF280 would have an eri0 internal and an hme in a PCI slot), but it is important that they have the same speed capability.

2. Obtain 4 IP addresses in the same local lan (or vlan) segment. In Multipathing there are 2 fixed (or private) address and 2 floating (or public) addresses. The 2 fixed addresses I refer to as internal. One is assigned directly to each hardware interface. The 2 floating addresses are the external ones. If one of the NICs detects link failure, the address tied to that NIC fails over to the working NIC. When the NIC comes back up, the address fails back to its original home. Determine right now which will be your internal IPs and which will be your external. I recommend keeping the same convention for all Multipathed machines, no matter what convention you choose. Here are two typical conventions:
   i. The first 2 IPs in the series are fixed and the second 2 are floating.
   ii. The odd IPs are fixed and the even IPs are floating (or vice versa)

3. edit **/etc/hosts** with your 4 IPS. example:

```
298.178.99.137    host-int0
298.178.99.138    host-int1
298.178.99.139    host-ext0 host-dummy
298.178.99.140    host-ext1 host.eng.auburn.edu
```

   In this example, the first two ips are fixed (internal) to the NICs, and the second 2 are floating. The last one is the one we use to tie to the machine name for programs that might have licensing restrictions tied to particular hostnames. (Always make the hostname tied to one of the public/external/failover NICs)

4. Configure network interfaces.

   At the beginning you'll have one network interface (the secondary) that is unconfigured, and another that would initially look something like this:

```
hme0: flags=1000843 mtu 1500 index 2
        inet 298.178.99.141 netmask fffffff0 broadcast 298.178.99.143
        ether 8:0:20:ff:5b:e2
```

   You need to configure the secondary interface and make it have a unique ether address that is persistent across reboots. I like to take the address of the hme0 (or eri0 or whatever) card and add 1 to the last octet.

```
# eeprom 'local-mac-address?=true'
# /sbin/ifconfig hme1 plumb
# /sbin/ifconfig hme1 ether 8:0:20:ff:5b:e3
```

5. Setup hostname.* files.

   You can pretty much copy these two files as is and just modify them slightly to fit your naming conventions in the same way that you setup the **/etc/hosts** file above.

   **/etc/hostname.hme0**

```
host-int0 netmask + broadcast + group production deprecated -failover up \
addif host-ext0 netmask + broadcast + failover up
```

**/etc/hostname.hme1**

```
host-int1 netmask + broadcast + group production deprecated -failover up \
addif host-ext1 netmask + broadcast + failover up
```

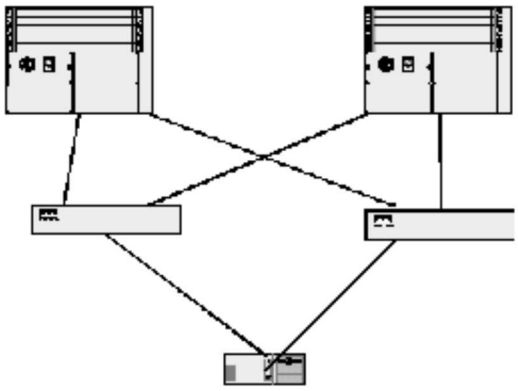
6. adjust failover detection timeouts

   **/etc/default/mpathd** has a default failover timeout of 10000. This means that it should take 10 at most seconds to detect and successfully fail over an interface. I like to configure this to 2500. In my working with IP multipathing, numbers below that seem to result in excessive messages about that number being too low and lots of messages in syslog. If you change this file, you will have to restart **mpathd**. Now is as good a time as any to either restart **mpathd** or start it for the first time if it is not already running.

7. If you use a default router, it must be pingable at all times from both interfaces. mpathd will ping your default router every at <FAILURE_DETECTION_TIME> second intervals. If you do not use a default router, then you need to run the router discovery daemon **/usr/sbin/in.rdisc**. This daemon should start automatically at boot time under the appropriate circumstances, but doesn't always (See Sun blueprint article for more thorough discussion). It is helpful to have a helper file to automatically start it if it is not already running. You can use this one if you like. Save it as **/etc/rc2.d/S70rdisc** and make a link in **/etc/init.d**

   When do you want to use which? It boils down to the same choices on a non multipathed host. Do you have one router on your lan or do you have multiple? If you have only one (or a pair using HSRP or other failover protocol), then you can use a default route. If you have more than one router, then you want to use **in.rdisc** much as you would use routed in a non multipathed host setup. Make sure you have router discovery announcements enabled on your routers in this situation.

   **TIP**
   Plug each physical interface into a separate switch to make effective use of multipathing. After that there are several ways you can configure your high availability. You can plug each switch into 2 routers and use HSRP to do router failover. In this case, having the Sun use a default route would be fine. Or, you could have each switch singly connected to a specific router on the same lan, and run **in.rdisc** on the sun to detect these interfaces and perform failover. A typical configuration is illustrated at right.



8. make it active

   This is the easy part. Copy and paste your /etc/hostname.hme* files to ifconfig commands as below:

```
# /sbin/ifconfig hme0 host-int0 netmask + broadcast + group production
deprecated -failover up \
addif host-ext0 netmask + broadcast + failover up

# /sbin/ifconfig hme1 host-int1 netmask + broadcast + group production
deprecated -failover up \
addif host-ext1 netmask + broadcast + failover up
```

9. Troubleshooting

   Occassionally you will see messages like this in your syslog files:

```
Nov 29 16:02:10 host.eng.auburn.edu in.mpathd[32]: [ID 398532 daemon.error]
Cannot meet requested failure detection time of 2500 ms on (inet  eri0) new
```

```
failure detection time is 5922 ms
Nov 29 16:12:29 host.eng.auburn.edu in.mpathd[32]: [ID 122137 daemon.error]
Improved failure detection time 3644 ms
Nov 29 16:12:29 host.eng.auburn.edu in.mpathd[32]: [ID 122137 daemon.error]
Improved failure detection time 2500 ms
```

I find that they are largely ignoreable. Failover still works.

There is a known issue with Solaris8 IMP where both interfaces can fail under high load if a particular patch is not installed. Reboot will not fix the situation, you must have the patch: 108528-15 (or later)

When you have a failure event of some kind, you'll see a message like this:

```
Nov 21 23:03:58 host.eng.auburn.edu in.mpathd[266]: [ID 832587 daemon.error]
Successfully failed over from NIC eri1 to NIC eri0
```

When it comes back, you'll see one like this:

```
Nov 23 15:25:00 host.eng.auburn.edu in.mpathd[266]: [ID 620804 daemon.error]
Successfully failed back to NIC eri0
```

If you see one like this, it's time to *run* to the switch closet:

```
Nov 23 15:23:56 host.eng.auburn.edu in.mpathd[266]: [ID 168056 daemon.error]
All Interfaces in group production have failed
```

---

Take the opportunity to test it out. Unplug one of your Cat5+ cables and watch failover work. Run a continuous ping to the machine. It's rather nice.

---

# Failover with 1 public IP

Now that you know how to setup resilient balancing links, you might be interested in how to setup a group with only 1 public, failover interface.

The advantages of this are

1. easier debugging - With the previous situation, you would have to **snoop** on both interfaces and correlate the traffic. With only 1 interface, you **snoop** in one place and see all traffic
2. easier firewalling - With 2 public interfaces, the traffic could initiate from either, possibly making firewalling a bit difficult since the source traffic could change from one IP to the other mid session.
3. 1 fewer IP consumed.

The following configuration has been tested and submitted by Eric Krohn

### Primary Interface

```
# cat /etc/hostname.hme0
DUMMY1  netmask + broadcast + \
group production deprecated -failover up \
addif REALNAME netmask + broadcast + failover up
```

### Standby Interface

```
# cat /etc/hostname.hme1
DUMMY2 netmask + broadcast + \
group production deprecated -failover standby up
```

**/etc/hosts file**

```
# cat /etc/hosts
#
# Internet host table
#
127.0.0.1       localhost
192.168.10.10   REALNAME  loghost
192.168.10.11   DUMMY1
192.168.10.12   DUMMY2
#
```

What does this do? It sets up two dummy (private) IP addresses that are fixed to the interfaces. It sets up a failover group named production. It adds an IP **REALNAME** to the group and marks it as the failover IP that will be migrated, and hme1 is set as the standby interface. In most situations, hme0 will be used to transmit and receive packets. In the case of failure (interface, switch, cable, router, etc), the IP for **REALNAME** will migrate to hme1 interface. When hme0 recovers, the IP will migrate back.

---

# References

Sun Infodoc 70062 discusses various permutations of IPMP configuration including multiple networks and no default router.

---

---

_mail pict_ Contact the author

Go to my homepage