



Hot Standby Routing Protocol

Peter J. Welcher

The last two months we looked at issues in Frame Relay network design and configuration. It's time we returned to LAN configuration. Let's take a look at the fairly new Hot Standby Routing Protocol (HSRP).

To understand the usefulness and cleverness of HSRP, however, we should first look at how hosts figure out which router to use to reach other networks or subnets.

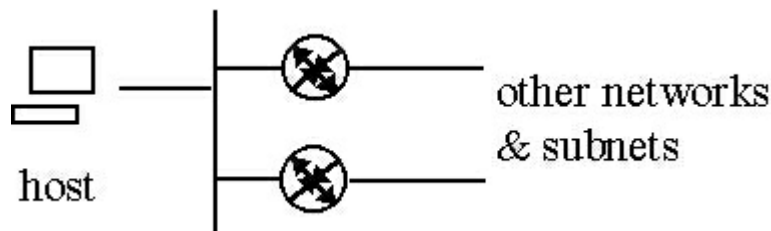


Figure 1. How hosts figure out which router to use.

There are several techniques, each with its advocates. They are:

- Run a routing process on the host
- Static default route(s)
- Proxy ARP
- GDP and IRDP
- HSRP

The first often seems desirable to high-powered workstation users. They want the best of everything! So routing gets fired up on their workstation, hopefully in passive mode, where the workstation listens but does not transmit. The workstation is certainly well-informed. But the CPU is being used to receive and process routing updates. This may somewhat decrease the computational performance of the workstation.

Worse is if the workstation or server has multiple interfaces. UNIX and OS/2 hosts may well think they're routers, so they'll transmit routing updates, and forward packets. Forwarding packets interrupts the CPU, generally a real performance hit. Routing updates aren't so great either, if the workstation isn't configured properly (wrong subnet mask?).

Router and networking staff generally think this is not good. Let routers do the routing!

Static default routes are the other extreme. PC's often can only be configured with 1 (to 3) default routers. Non-local traffic is then sent to the default router. This is simple, low-impact on the host, but in case of an inoperative router or link, may not allow an alternative router or link to be used, depending on the implementation. UNIX workstations may be booted with multiple default routes, but may not recover the routes if gateway routers become unavailable.

Proxy ARP is common in PC environments too. The idea is to convince the PC or host to ARP for all IP addresses as if they were on the local wire. How we do this depends on the implementation. A router then responds to the ARP request with its own MAC address, and the host then sends packets to that MAC address. The router forwards the packet to the destination, and the host never realizes it didn't have the MAC address of the destination.

The only problem with proxy ARP is that the subnet mask configured into the PC's may be somewhat odd, like 0.0.0.0. This says that no part of the address is significant for routing decisions, or all addresses are on the local wire. That certainly strikes me as odd! It also puts extra traffic on the local link, whenever the host wishes to send packets to a new IP address. The host sends out an ARP broadcast, and all the local routers respond. So for each new destination there's a broadcast and one or more responses, all bearing the same MAC addresses as the previous time. Broadcasts interrupt every CPU on the link, so this doesn't scale up that well, not to mention being somewhat unnecessarily chatty.

Gateway Discovery Protocol, due to Cisco, is an alternative. The idea is that routers broadcast periodically (every 15 to 30 minutes usually), announcing they're open for business, and that this information is valid for a certain period of time. With multiple routers, priorities can be announced, giving preference to one router over another. Hosts that boot can solicit information (one time only).

ICMP Router Discovery Protocol (IRDP) cleans this up to use multicast, and is described in RFC 1256. Cisco routers support both GDP and IRDP.

Why doesn't the world use IRDP or GDP? Well, you need code running on your host. Code is available on [ftp.cisco.com](ftp://ftp.cisco.com/pub/rdisc) in `pub/rdisc`. IRDP is also available for Suns at some of the usual sites, as `irdpd.c`. All you have to do is compile and you've got a process for your Sun that updates the default route in the routing table whenever necessary. If you've got another flavor of Operating System, well, you get to port the code.

I imagine that when Cisco looked at the support issues here, porting code, supporting various platforms, the user-installation issues, it might not have seemed very attractive. And from all this came HSRP, a nifty trick that dodges many of the issues that come up with other approaches.

The idea behind HSRP is to establish a virtual router (with its own IP address) as the default router for the hosts on a LAN. The virtual router also gets its own MAC address. One or more routers then pool as the standby group for this virtual router. One of the routers in the pool is active at any time, actually forwarding packets sent to the virtual router's MAC address. If that active router disappears, another router in the pool takes over. The advantage is that the host computer never knows that different routers are involved. It just sends packets to the virtual router, oblivious to the actual router that forwards the packets. And it only has to ARP once, to get the MAC address associated with the virtual router's IP address. So this saves all the ARP traffic that comes with proxy ARP. It also accomodates host implementations that ignore ARP table changes, a problem with moving a MAC address from one

IP address to another (one real router's address to another's).

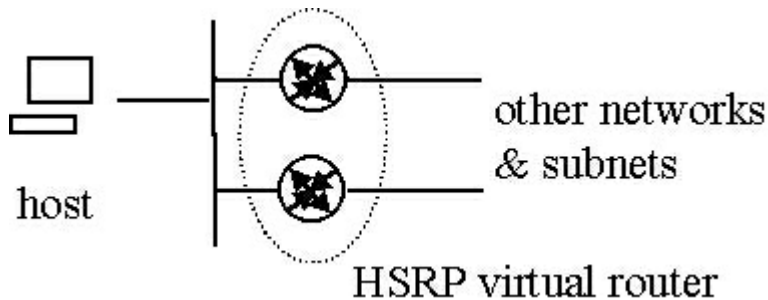


Figure 2. HSRP uses a virtual router.

Configuring HSRP is easy. All we configure is

```
interface ethernet 0
ip address 131.108.1.1 255.255.255.0
standby 2 ip 131.108.1.3
```

On the second router attached to the Ethernet LAN:

```
interface ethernet 0
ip address 131.108.1.2 255.255.255.0
standby 2 ip 131.108.1.3
```

This puts both routers interfaces in the same subnet, with a common standby group of 2 on that link. So both routers are responsible for acting together as the virtual router 131.108.1.3. Hosts are configured with a static default gateway, IP address that of the virtual router, 131.108.1.3.

And that's all it takes!

The HSRP can be used on LAN interfaces: FDDI, Token Ring, and Ethernet. On some router models and interfaces you can use multiple HSRP groups. One use might be to create two virtual routers, the groups spanning two actual routers. Point half of the LAN hosts at one virtual router, half at the other. Use different priorities (see below) so one actual router is active as the first virtual router, the other as the second virtual router. This load balances, and if either router dies, the other one takes over for it.

There is a limitation of at most 3 standby groups (virtual routers) per Token Ring interface. The 1000, 2500, 3000 and 4000 model routers with Lance Ethernet chips can only support one Hot Standby group per Ethernet interface. I assume this is primarily a hardware limitation: you don't really want the CPU of the router looking at every routed packet, so the interface has to provide some selectivity as to the MAC addresses it interrupts the router CPU with.

The underlying mechanism is UDP multicasts: the routers periodically send group HELLO messages to let their peers know they are still there.

Once you've got this basic idea, the other HSRP commands are nerd knobs allowing you to tweak the settings. By the way, the HSRP commands are all interface commands.

The command

```
standby 2 timers 1 3
```

sets the hello and hold timers for standby group 2. These are the default values of 1 second between hellos and 3 seconds before assuming a router is down.

To control which router is active, configure

```
standby 2 priority 90
```

The default priority is 100, higher priority wins.

To allow a router to resume being the active router for group 2, add

```
standby 2 preempt
```

There is also a command that lets you track interfaces and lower the priority if any of the interfaces is down (making the router less desirable as a default gateway). The default priority increment is 10, but you can configure other increments. Increments other than 10 are cumulative. So if several interfaces are down, the configured increments are all subtracted from the priority level of the router. Here's what the command looks like:

```
standby 2 track ethernet 0 25
```

To monitor standby, we can use the commands we'd expect:

```
show standby
```

and

```
debug standby
```

It's that easy!

Dr. Peter J. Welcher (CCIE #1773, CCSI #94014) is a Senior Consultant with Chesapeake NetCraftsmen. NetCraftsmen is a high-end consulting firm and Cisco Premier Partner dedicated to quality consulting and knowledge transfer. NetCraftsmen has nine CCIE's, with expertise including large network high-availability routing/switching and design, VoIP, QoS, MPLS, network management, security, IP multicast, and other areas. See <http://www.netcraftsmen.net> for more information about NetCraftsmen. Pete's links start at <http://www.netcraftsmen.net/welcher> . New articles will be posted under the Articles link. Questions, suggestions for articles, etc. can be sent to pjw@netcraftsmen.net .
