How to Implement OSPF

RIP can cause problems on all but the smallest networks. Open Shortest Path First was developed by Radia Pearlman and reduces the incidence of needless traffic.

By Neil Briscoe
Network Consultant

On the whole, networks in any company, even reasonably small ones, are getting larger and more complicated. As they do so, some things on the network begin to suffer from the stresses this entails. Not least of these are the routing protocols used in your network. For small networks, RIP (the Routing Information Protocol) is just fine; it works and, best of all, all the devices on your network understand it. Unix, NT and NetWare all understand RIP, almost "out of the box", although there is usually a small amount of configuration to be done.

RIP works by sending routing updates every 90 seconds, even if there have been no changes to the routing topology (which, after all, tends to be the norm). As the number of routers and associated routes increases, these routing updates take up more and more of the bandwidth. Furthermore, because every device on the network understands RIP, more and more of their processor time is taken up "just" dealing with the routing updates, which are often no such thing in any event, so simply serve to waste time. The user of such a network will simply notice that things appear to be congested, and shortly they will be on the phone to you asking if you can speed things up.

Given that network topology changes are conveniently rare (management have to spend money to change them) it would be nice if we could use a routing protocol that only sent out updates when there was an update to talk about. And, given the new Access Server on the network, the routing updates are going to be intolerable once users are dialling up and adding, and then removing (as they disconnect), routes between the network and themselves. Servers will crash under the load if we continue to use RIP.

A New Protocol

Fortunately, this was all foreseen some time ago, and Radia Pearlman spent a great deal of time developing the protocol that is now known as OSPF (Open Shortest Path First).

Let us assume that we have a functional and currently quiescent network topology, and that all the routers have a sane routing table. In such a state, we don't really want to flood the network with a routing update which simply restates our view of the world. So a router will content itself by sending out an OSPF HELO, which simply serves to tell its peers that it is alive and well. Given that it issues no routing updates, all of its peers feel free to assume that the topology they "know" remains the same.

Now, let us assume that we suffer a link failure on the connection to network X. The router directly responsible for the link notes that it has gone down, since it notices the DTR on its interface go down. Immediately, it multicasts a message to all its peers telling them that it has lost a route and should no longer be considered as being able to talk to network X.

The important point here is that the responsible router only tells its peers about the one route loss, not also the remainder of its routing table. Its peers update their tables accordingly, removing that route from their tables. After the event, all routers have a sane, consistent routing table - this time lacking a route to network X. They will not have to update their tables again until such time as the

router that has a link to network X notices its DTR go high once more, and multicasts that fact to its peers.

Access Servers

Now consider an Access Server, which typically has lots of routes which are considerably less static. They "flap" all the time as users dial in, do their thing, and disconnect. ("Flap" is a term used by router technicians to designate a route that appears and disappears more often than it should. For a fixed link a "flapping" route is a bad thing, since it causes routers to issue and receive more routing table updates than necessary. It is, however, in the nature of dial-up connections that routes transition frequently.)

Most servers don't understand the OSPF protocol but, nevertheless, will be affected if the LAN they are on is congested by excessive multicast packets from an OSPF router. Placing an Access Server on the backbone, therefore, is a design decision which you will later need to re-evaluate.

Unix servers can be made to understand OSPF by installing gates on them. NetWare 5, whilst not running OSPF "out of the box", can certainly be made to utilise the protocol. This author is unaware of any means of coercing NT servers to do the same. However, a good core network design renders it unnecessary to ask any server, Unix or otherwise, to deal with routing packets. That is, after all, the job of a router, which is specifically designed to handle the routing of network packets rather than serve data or run applications.

What you can do is to place a buffering router on the backbone. This router will have at least two LAN interfaces. One of these will be on the backbone; the other will be on a separate LAN to which you then connect your Access Server(s). Now, as users dial in and disconnect, the Access Servers will only multicast on their own LAN, keeping the routing updates to that locality and not overburdening the servers on the core network. Of course, the core needs routes back to the users. However, the buffering router can offer what is, in effect, a static route to the core that rarely if ever changes, keeping routing updates to a minimum.

Kiosks

The buffering router can do this only if IP addresses are chosen with care. Imagine a situation where "kiosks" will dial into an Access Network. Each kiosk consists of an ISDN router and a computer running NT. On the kiosk's LAN, therefore, you need just two IP addresses - one for the Ethernet address for the router, and one for the NT machine. This mandates a network mask of 255.255.255.252, which allows for just two usable addresses. If our first kiosk uses 192.168.2.0/30 (giving it usable addresses of 192.168.1.1 and 192.168.1.2), our second kiosk can use 192.168.2.4/30, our third 192.168.2.8/30 and so on.

"Most servers don't understand the OSPF protocol but, nevertheless, will be affected if the LAN they are on is congested by excessive multicast packets from an OSPF router."

This clearly allows the buffering router to offer a summary route of 192.168.2.0/24 to the core, and this doesn't need to change, no matter how often kiosks connect and disconnect. Now our core servers and routers are unaffected by changes that go on at the periphery. In my example, I've used a Class C address to illustrate how you might design a small kiosk network. However, clearly, for anything major, a much larger addressing space is required. This author uses a subnetted

```
hostname buffer1-gw
interface FastEthernet0/0 ip address 10.1.0.1 255.255.0.0
no ip redirect
interface FastEthernet0/1 ip address 192.168.1.1
255.255.255.0
router ospf 1 redistribute connected metric 6 subnet
redistribute static metric 6 subnet network 10.1.0.0
0.0.255.255 area 0 network 192.168.1.0 0.0.0.255 area 1
default-metric 6
```

Figure 1 - A partial configuration for a buffering router.

Class A address space to provide services for a kiosk network.

Finally, in our kiosk dial-up design, we decide that the dialer interfaces on the kiosk routers (the WAN interfaces) will be in their own network. So that we can use multiple blocks of 192.168.x.y addresses for kiosk LANS as their number increase, we use a class B address for the dialer interfaces. We'll choose to use 10.2.0.0/16 as the WAN network. The kiosk router with the LAN address 192.168.2.1 will have 10.2.2.1 as its dialer address. The next router will use 10.2.2.5 etc.

This does lead to some wastage of addresses; however, some of these will be used by dialer interfaces on our Access Servers. We may have an access server with lines from multiple telephone companies, or which otherwise form separate dialer groups. In addition, if things become really large, we may require multiple Access Servers to handle the number of lines. Each separate dialer interface, whether on just one or multiple chassis, will need to have a separate dialer interface defined, and hence a separate IP address. We can use any of the unused IP addresses on the 10.2.0.0 network for this purpose.

To allow for multiple Access Servers, even if we only have one now, we reserve the 192.168.1.0/24 address block for the Access Server network. We give the buffering router an address of 192.168.1.1 on this network, and the first Access Server gets 192.168.1.2. The buffering router also gets an address on the core network. We'll assume we're using 10.1.0.0/16 for this example and our Access Server has an address of 10.1.0.1.

Backbone Area

OSPF works on the basis of having a backbone area, known as area 0. It also has the concept of Autonomous Systems. All areas within an autonomous system can only communicate with each other by transmitting the backbone. Separate autonomous systems need to have their backbones adjacent to each other.

In our network, clearly the core network will be the backbone area. We will place the Access Server network in area 1. We cannot, therefore, define an area for the temporarily connected kiosk networks, since they are now multiple hops from the backbone. We can, however, have OSPF redistribute our static and connected routes. A static route is one you manually enter into the router configuration; a connected one is formed when you apply an address to an interface, indicating to the router that it is connected to that network.

Dial-up networks such as the one described normally have the static routes applied as users connect, and removed as they disconnect. This task is carried out by the associated RADIUS or TACACS server used on the network, and will not be described here.

"The routing updates are going to be intolerable once users are dialling up and adding, and then removing, routes between the network and themselves. Servers will crash under the load."

Configuration Example

Figure 1 shows a partial configuration for our buffering router (I've only indicated the parts that relate to interfaces and OSPF itself). This shows that we have numbered our interfaces. The "no ip redirect" on the backbone interface is there

```
hostname access1-1-gw
isdn switch-type primary-net5
interface serial0:15 no ip address dialer-group 1
encapsulation ppp isdn switch-type primary-net 5
interface dialer 1 ip address 10.2.2.2 255.255.0.0
encapsulation ppp ppp authentication chap chap hostname
access1
router ospf 1 redistribute static metric 6 subnet
redistribute connected metric 6 subnet network
192.168.1.0 0.0.0.255 area 1 passive-interface Serial0:15
default-metric 6
```

Figure 2 - Partial configuration for Access Server.

because, if you have other routers on the backbone which also run OSPF, our buffering router will have learnt routes to other networks. By default, when a machine using the buffering router as a default gateway sends a packet to it that is served by another router in the core, it will issue an ICMP redirect. The "no ip redirect" command turns off this default behaviour.

A note on ICMP redirects: an ICMP redirect packet will be issued by a router when it receives a packet from a device on the same network as the interface on which it receives it, and when the destination address is served by a third machine on that same network. It says the equivalent of: "Don't talk to me, talk to that machine over there - it has a shorter route". Many network servers simply don't handle ICMP redirects gracefully, especially if you've configured them not to run routing protocols.

We chose to make the ASN for our OSPF grouping 1. We could have picked any number within the supported range. The next commands cause our OSPF router to redistribute any static and connected routes of which it is aware. The network commands define our areas, telling the router which IP addresses fall within the backbone and which within an adjacent area. By using the two network commands, the buffer router has become an "Area Border Router". Note that what appears to be a very strange pair of subnet masks are not subnet masks at all but compare bits. If you look closely, you'll see that, for each octet, the value is formed by subtracting the subnet value from 255.

On our Access Server, a partial configuration might look as shown in Figure 2. In our configuration, we have assigned the first ISDN30 channel to a dialer group, and then assigned an address to the dialer group. This allows us to then add additional ISDN30 channels into the same dialer group as we expand. In order to facilitate this, we define a chap hostname to the dialer. We can then use this chap hostname on multiple dialers and multiple chassis, allowing our kiosks to dial any of our numbers until they obtain a connection.

Our OSPF block tells this router it is an Area Boundary Router falling within area 1. It directly exchanges routes only with the border router, and any other Access Servers within this area. However, if the border router picks up routes from other routers within area 0, it will exchange these with the Access Servers, thereby allowing the Access Servers to route packets from connecting kiosks to anywhere on the network, and our goal has been achieved.

A useful command for checking routing on the routers is "show ip route", which will show all routes picked up by running any routing protocol. You'll be able to see which routes are local to the area(s) in which the router lies, which are Inter Area routes, and which are external routes. These latter tend to be the static and connected routes that we redistribute using the protocol. Another is "show ip ospf", which will tell you how many times the SPF protocol has been run.

PCNA

"For small networks, RIP (the Routing Information Protocol) is just fine; it works and, best of all, all the devices on your network understand it."

New Reviews from Tech Support Alert

Anti-Trojan Software Reviews

A detailed review of six of the best anti trojan software programs. Two products were impressive with a clear gap between these and other contenders in their ability to detect and remove dangerous modern trojans.

Inkjet Printer Cartridge Suppliers

Everyone gets inundated by hundreds of ads for inkjet printer cartridges, all claiming to be the cheapest or best. But which vendor do you believe? Our editors decided to put them to the test by anonymously buying printer cartridges and testing them in our office inkjet printers. Many suppliers disappointed but we came up with several web sites that offer good quality cheap inkjet cartridges with impressive customer service.

Windows Backup Software

In this review we looked at 18 different backup software products for home or SOHO use. In the end we could only recommend six though only two were good enough to get our “Editor’s Choice” award

The 46 Best Freeware Programs

There are many free utilities that perform as well or better than expensive commercial products. Our Editor Ian Richards picks out his selection of the very best freeware programs and he comes up with some real gems.