



# Introducing OSPF

**Peter J. Welcher**

---

## Introduction

The OSPF routing protocol is playing an increasingly important role in modern networks. It is a multi-vendor standard. We'll take a look at OSPF basics in this article.

While I and a number of people are fans of Cisco's EIGRP for its simplicity and ease-of-use, lately I'm seeing more use of OSPF. This may be due to networking staffs having increasing levels of sophistication. Or it may be that folks are looking at to the next generation of hardware, and thinking that OSPF is something you'll find in any worthwhile Layer 3 switch ("fast router"). After all, who'd want to buy a fast Layer 3 switch that only routes using RIP or RIPv2, with their slow convergence? That's basically the "it's a standard" argument -- you do OSPF if you anticipate using mixed vendor equipment in your network, or if you at least want to maintain that possibility.

Running OSPF so your routers can exchange information with NT servers is probably **not** a Good Thing. I personally don't want servers sending routing information to my "official routing protocol", since I've seen too many servers with duplicate IP addresses or other oddities. If a server administrator goofs and duplicates the server farm subnet on some workgroup server somewhere else, and it advertises it into your network, you're in for a serious outage! If your group controls the servers, or if NT servers are acting as low-end routers at remote sites, well, then there's a reason to run OSPF. I'll just mention in passing here that OSPF has an authentication feature that allows you to easily exclude any "unofficial" routers from your OSPF Autonomous System ("group of consenting routers").

I don't think I'm revealing any deep dark secret if I tell you that you need to know OSPF well for the CCIE test and for recertification. The general rule of thumb is, if it's a major protocol or feature, used at a lot of sites, it's probably in the test. (There's also minor stuff, presumably to see if you can Read The Fine Manual and get it working).

## Pros and Cons of OSPF

The chief reasons for using OSPF are fast convergence and the fact that it's an IETF standard. It does minimize bandwidth consumed by routing information, using low bandwidth wisely, and it does take WAN link bandwidth into account, preferring high bandwidth paths.

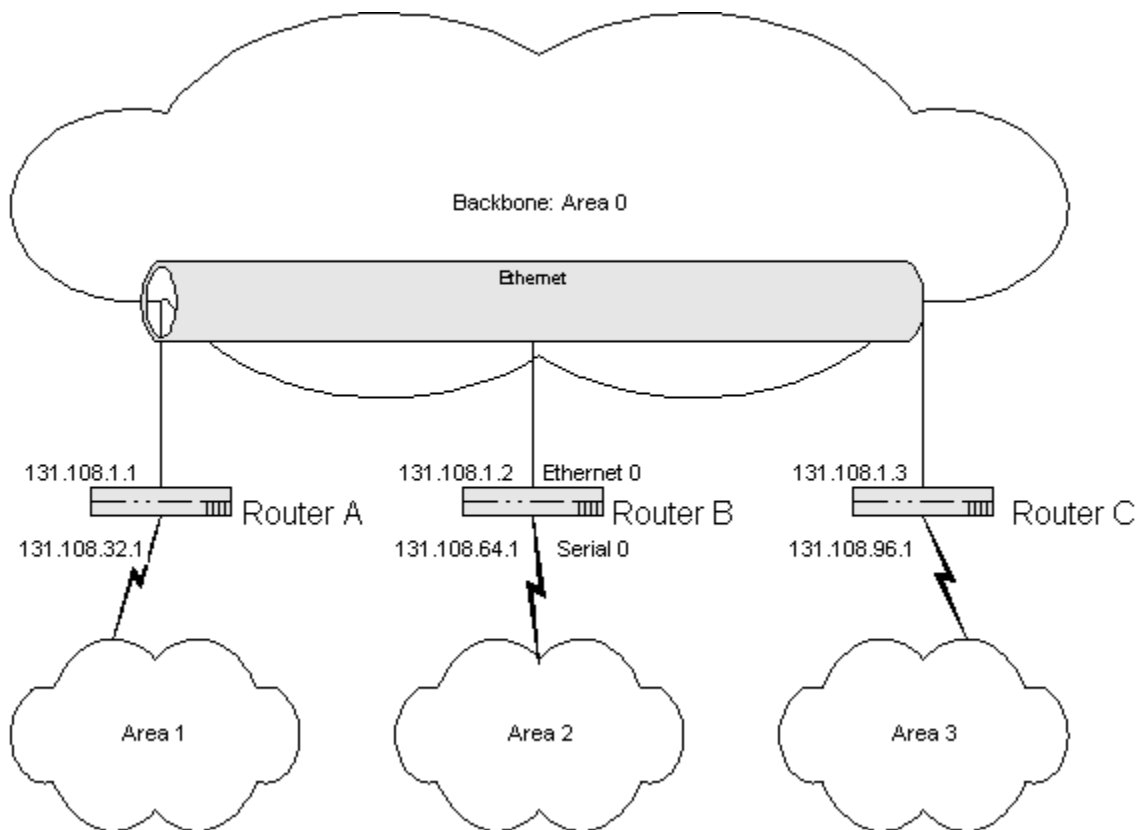
The drawbacks are the area structure and design aspects of OSPF, which we'll look at below. This

means you have to plan for how your network is going to grow when doing an OSPF design -- and some people would say that's a good thing. The protocol is mildly complex, so there is perhaps a steeper learning curve than EIGRP. Finally, configuring OSPF is a little more work than configuring EIGRP.

## OSPF Design

Before we can do much else, we need to talk a little bit about OSPF design. OSPF requires that you divide your network up into a logical star of areas (also sometimes referred to as the "daisy" or "black-eyed susan" model, since it looks somewhat like how children draw flowers). Areas are supposed to be contiguous (all in one connected piece). However, if links fail and render areas other than area 0 discontinuous, OSPF has a mechanism for handling this. The following figure shows this topology, where an Ethernet segment (and perhaps some other network links) constitute area 0.

OSPF must have a backbone area 0. Any other areas must connect to area 0. All traffic between areas must go through area 0. Think about that. That implies that Area 0 is pretty critical to your network. It had better be robust and stable. It should probably have internal redundancy. And it better have the bandwidth to handle the traffic between areas.



There are some rules of thumb for Cisco router OSPF designs:

- A router should be in at most 3 areas (and watch your CPU and RAM).
- An area should contain 50 to 100 routers (I've also seen 25 in some Cisco documents).
- A router should have fewer than 60 neighbors.

These not hard and fast guidelines, but they tell you when you're perhaps stretching things a bit.

So if your network has fewer than 50 to 100 routers, you may be able to just have one big (happy?) area 0.

When you try a real OSPF design, you'll find that OSPF area 0 tends to grow on you. I've been calling this the "vacuum" effect: OSPF area 0 almost seems to pull links into it, until you find your design is all one big area 0, and you have to start over. My advice is to keep OSPF area 0 small. If area 0 is small, it's easy to keep it robust. I prefer to put the links connecting to non-backbone areas into those areas, and keep them out of area 0 (possible exception: dial backup links).

Terminology: routers within area 0 are **backbone routers**. The routers connecting area 0 to other areas are **Area Border Routers** (ABR's). Routers purely within an area are **internal routers**. And routers that redistribute another protocol into OSPF, or that connect to another routing domain, are **Autonomous System Boundary Routers** (ASBR's).

The routing pattern in OSPF seems to occasionally surprise people in more complex designs, so let's repeat how traffic goes between areas. Traffic from Area 1 to Area 2 must go through area 1 until it reaches an Area 1 Border Router. It then travels only within the backbone Area 0 until it reaches an Area 2 Border Router. It then travels within Area 2 to the destination. Traffic to a destination within the same area as the source travels purely within that area; it may not normally cross into the backbone.

The reason for areas is the old computer science trick of "divide and conquer". OSPF routers in an area share all information about routers and link status. This uses RAM, and computing the Shortest Path First (Dijkstra) algorithm can use CPU. Furthermore, routers in an area share information via Link State Advertisements (LSA's), which are flooded throughout the area. (Chicken and egg problem: routers may not know the other routers in the area or how to get to them, so LSA's have to slosh -- umm, flood -- around to make sure every router gets a copy). By dividing the network into areas, the amounts of RAM, CPU, and flooding traffic are limited, providing better scalability.

We'll also see that we can summarize routes at Area Boundary Routers (ABR's), hiding the details of the area, and in effect, advertising "this way to the area". That's good for stability and scaling of the network.

Routers within an OSPF area must share the same information (so as to derive the same map of the area and the same routing conclusions). Thus you cannot filter or hide routes within an area.

## Configuring OSPF

Let's take a look at how Router B in the middle of the figure might be configured. Note that it connects to two areas, so it has at least one interface in each area (the drawing doesn't show more, because it's cluttered enough the way it is).

I've already assigned addresses that might be summarizable. The design for this network is to divide 131.108.0.0 up into 1024 subnets using subnet mask 255.255.255.192. Thus the subnets will have some number in the third octet, and will have multiples of 64 in the last octet (some companies call this "quads"). (Calculating  $256 - 192 = 64$ , we see we have multiples of 64 in the fourth octet, but the third octet is also part of the subnet information.)

The subnets are assigned in blocks, as follows:

Area	First Subnet	Last Subnet
0	131.108.0.0	131.108.31.192
1	131.108.32.0	131.108.63.192
2	131.108.64.0	131.108.95.192
3	131.108.96.0	131.108.127.192

As we may see in a future article, this makes them summarizable blocks. If you're into bits and binary, you can think in terms of an "area mask" of 255.255.224.0. All the subnets in each block above, when logically ANDed with this area mask, share the same bits. Another way to think of this is to calculate  $256 - 224 = 32$ , so with mask 255.255.224.0 we're working with multiples of 32 in the third octet (where the 224 was).

Note that the Ethernet 0 addresses are all drawn from a subnet in the block of subnets for area 0. And the serial 0 interfaces come from the first subnet in the block for their area.

Here's the relevant router configuration:

```
interface Ethernet 0
ip address 131.108.1.2 255.255.255.192

interface Serial 0
ip address 131.108.64.1 255.255.255.192

router ospf 1
network 131.108.0.0 0.0.31.255 area 0
network 131.108.64.0 0.0.31.255 area 2

area 0 range 131.108.0.0 255.255.224.0
area 2 range 131.108.64.0 255.255.224.0
```

The interface command show that we assign addresses as usual, using normal subnet masks. (In a real network, we would probably do Variable Length Subnet Masks -- VLSM -- on Serial 0, with a mask of 255.255.255.252, but I've chosen to keep the example simple).

The router command specifies an OSPF routing process with process id 1. This number is not part of the OSPF LSA's, and hence is purely local to the router. You should probably use 1 to make it easy to remember, unless you have multiple OSPF AS's, in which case you might choose to use the process id to track which OSPF process goes with which OSPF AS.

The network command for OSPF does a couple of things. It tells the router which interfaces are in which areas. And it causes the router to start sending OSPF hellos on matching interfaces. The gotcha with the network command is that it uses a wildcard style mask. In the above example, the mask indicates that any interfaces in the range 131.108.0.0 through 131.108.31.255 should be in area 0, and interfaces in the range 131.108.64.0 through 131.108.95.255 should be in area 2.

```
131.108.  0.  0      131.108. 64.  0
   0.  0. 31.255    0.  0. 31.255
-----
131.108. 31.255    131.108. 95.255
```

The nice thing here is that you can actually add the mask to the starting address to find the end of the

range of addresses covered. This only works if the block or starting address is correct for that mask (I'm going to be a bit vague here, to duck having to do more binary arithmetic in public.)

Some people prefer to Keep It Simple, configuring `network 131.108.1.2 0.0.0.0 area 0`. This tells the router that precisely the address 131.108.1.2 is in area 0. That's certainly simple: keep the mask 0.0.0.0, and just specify each interface address. Of course, it's a bunch of typing on a bigger router with lots of interfaces.

By the way, when you're experimenting with OSPF, there can be order sensitivity to the commands. The problem seems to be that if you configure `network ...`, then `no network ...`, then `network <something else>`, the router may not re-initialize OSPF on the relevant interface(s) -- so it doesn't send hellos, and doesn't form adjacencies. The solution is either to save your configuration and reboot the router (drastic), or use a mouse to copy the router ospf part of the config, do `"no router ospf 1"` to stop OSPF, then paste the correct config back in.

To troubleshoot OSPF, the command to begin with is

```
show ip ospf neighbor
```

The use of this is simple: if the routers aren't fully adjacent (neighbors), they're not exchanging information. We'll look more at what the show command and debug can tell you, as well as router id's, adjacencies, hellos, LSA's, and how OSPF works, in a future article.

## OSPF Design, Revisited

Suppose you have an OSPF network, and your company merges or acquires or needs to communicate with another that also runs OSPF. You could try to glue the area 0 backbones together, but Murphy's Law says they'll be situated so as to make that real fun. Another possibility is to use multiple OSPF Autonomous Systems (routing domains), and run some other protocol (BGP?) between the Autonomous Systems (AS's).

This may also be useful in a multi-national corporate network. Suppose your company has networks in the North America, Europe, Asia, Australia, and perhaps those other continents too. Are they under common administration? One suspects not. So controlling growth, addressing, and so on centrally could be a challenge. Instead, you might consider making each continent a separate OSPF network, using some other protocol to control the flow of routes or summary routes between the continents. The granularity here doesn't have to be continents, I'm just using them to make the point.

The U.S. Post Office is putting in 38,000 Cisco routers. Note that if we use our rules of thumb, 38,000 divided by 100 routers says we'd need 380 areas. Alternatives include multiple OSPF AS's, and other protocols (like EIGRP). I'd very definitely want healthy route summarization in such a network. Note that since OSPF only allows us to summarize at ABR's, we can only have one level of summarization. With 38,000 routers, we might well want to summarize at the regional, state, and portion of state levels. That can't be done with pure OSPF.

## Conclusion

We haven't had room to talk adequately about route summarization and why you want it in larger networks. That sounds like a future article! In that article, we'll see that route summarization plays a

role in controlling the effects of a flapping link. We'll also talk about stub areas, which are a good way to make your OSPF designs more stable as well. And we'll talk about some of the advanced features of OSPF.

Did I miss a good reason for running OSPF? Are NT server router wannabe's a problem in your network -- or a solution? Is there something else you'd like to know? Let me know by email! And while you're at it, ideas and wishes for future columns are always welcome!

---

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](mailto:pjw@netcraftsmen.net) .

---

9/9/98

Copyright 1998, Peter J. Welcher