



# OSPF and Route Summarization

**Peter J. Welcher**

---

## Introduction

Two months back we looked at basic configuration of OSPF routing on Cisco routers. Last month we looked at summarizable ranges of addresses. We saw that summarizable blocks of addresses consist of some number  $N$  of addresses, where  $N$  is a power of two, and where the key octet of the first address in the block is a multiple of that number  $N$ .

If you missed either article, please look for them at <http://www.netcraftsmen.net/welcher/papers/index.htm>, along with other previous articles.

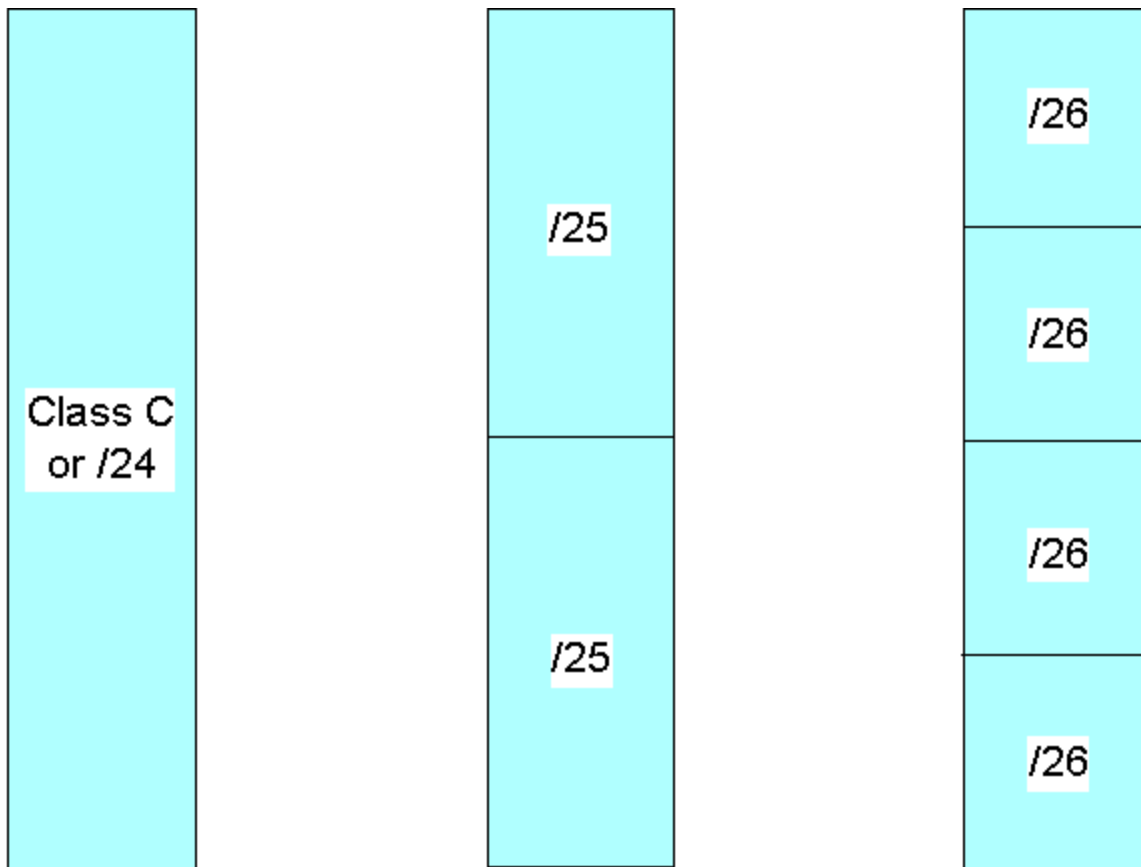
This month, we'll see how to think of summarizable address blocks in a slightly different way. And then we'll see how this all applies to configuring route summarization with OSPF.

## Thinking About How Big Blocks of Addresses Are

Let's start by getting our bearings as far as size of address blocks. A Class C network address is a /24, with subnet mask 255.255.255.0. Let's draw that as a "basic sized" rectangle, as shown in the figure.

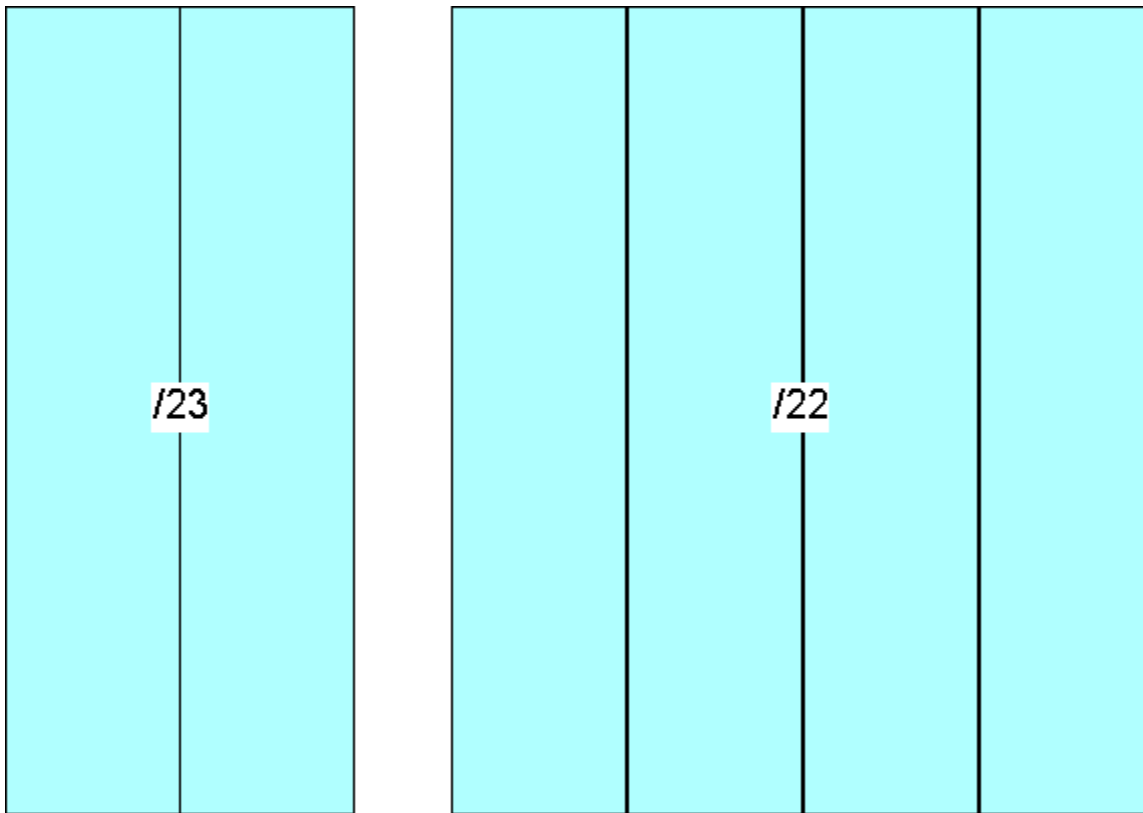
If we make the mask a /25, which is subnet mask 255.255.255.128, the subnets are multiples of 128 in the last octet, namely 0 and 128. This is legal with classless routing protocols. Note that this cuts the Class C address space into two halves, as shown in the figure.

If we make the subnet mask a /26, or 255.255.255.192, the subnets are multiples of  $256-192 = 64$ , namely 0, 64, 128, and 192. (Some folks call this "quads", when you divide a Class B into /26's: the subnets of such a class B are some number in the third octet and 0, 64, 128, or 192 in the fourth octet). With quads, we've cut the Class C address space into quarters, as shown in the figure.



So increasing the number of bits **decreases** the size of the subnet. Adding one bit of subnet mask **halves** the address space.

What about going the other way? Well, if adding one to the number of subnet bits halves the address space, then subtracting one does what? It **doubles** the address space! That means that a /23 is two Class C's worth of address space, for example 200.1.10.0 and 200.1.11.0. And a /22 is four Class C's worth, for example 200.1.20.0, 200.1.21.0, 200.1.22.0, and 200.1.23.0. Of course, these might be instead drawn from some larger block: 10.50.20.0 through 10.50.23.0. See the next figure.



## Longest Prefix Matching

Recall that classless routing protocols are those that pass around the subnet mask in routing updates, instead of assuming they know it. (They don't assume they know the subnet mask from appearances). Classless routing protocols include EIGRP, OSPF, RIPv2, IS-IS for IP, and BGP.

I like to think of routing table entries as telling the router how to deliver packets to **blocks of addresses**.

Including subnet mask information just clues the router in on how big the block of addresses is. That way, the router can tell the difference between 10.50.20.0 /24 and 10.50.20.0 /22. The former is a Class C-sized block of addresses, 256 addresses. The latter is four Class C sized blocks, or 1024 addresses. Of course, host computers and router interfaces cannot be given addresses at the ends, corresponding to the name of the subnet and the directed broadcast address.

When there's a choice, the router uses the most specific information available. My example is that when you're driving and you see a highway sign saying New York **State**, pointing in one direction, and New York **City**, pointing in another, you're going to follow the most specific sign. If you're far away, it probably doesn't matter. But if you're in Pennsylvania or New Jersey, headed for New York City, you'd better follow the sign for the city. If you're headed for New York state, well you'd better know more, like what roads go to what parts of the state. (Also, some of the state thinks Albany is upstate, the rest thinks it is central, but that's another matter!)

Similarly, suppose a router has a packet bound for 10.50.20.10, and has routes to 10.50.20.0 /24, 10.50.20.0 /22, 10.50.0.0 /16, and 10.0.0.0 /8. Then the router should use the route for 10.50.20.0 /24,

the **longest prefix match**, the smallest (most precise) address block that matches the packet's destination. If the packet is headed for 10.50.21.30, then the 10.50.20.0 /22 route is the best match. If the packet is headed for 10.50.40.10, then the 10.50.0.0 /16 is the best match. And if the destination is 10.60.1.1, then the 10.0.0.0 /8 is the best match in this (rather short) routing table! (And, like the sign for New York State, it may not be a very optimal route).

## OSPF Route Summarization

To follow the road sign analogy a little more, imagine that the highway sign listed all possible streets in New York City, vice the summary "New York City". You'd have to stop for a while, just in order to read it. Luckily, Cisco routers can access the routing table faster than that (i.e. it probably isn't a linear search for a routing prefix match). You can also imagine the highway department posting corrections, like putting "Closed for Repairs" next to the 100th Street listing.

To see the significance of that last remark, bear in mind that a subnet route is listed in the routing table when the associated interface is up. If the interface is down, then the route is not advertised to neighbors. (Or, in OSPF, the link is down, so no subnet route is computed). If your routing tracks your network, subnet by subnet, you are indirectly tracking the up/down state of every LAN and WAN link. Hence the analogy about updating the road sign to reflect construction.

Route summarization therefore has two benefits. There is the more obvious benefit, of shrinking routing tables. With distance vector protocols, this includes the benefit of reducing routing update traffic. The less obvious benefit is that summarization means you're tracking whether or not you're connected to some subnets of a summary, not the up/down state of every link. Thus when the link goes up or down, you don't have a flurry of traffic announcing the state change. (OSPF and EIGRP both react quickly to changes, the drawback being that one might consider them to be a bit hyperactive.) And when you have a flapping interface (up/down over and over again), this can make a real difference. With OSPF, recalculating the Dijkstra algorithm can take significant CPU and time (a couple of seconds, anyway) -- doing this over and over again in response to a flap is a bit ugly.

One design situation where route summarization is important stands out. Dial interfaces are intended to flap! As calls come in and later terminate, the dial interfaces go up and down constantly. Summarization isolates the rest of your network from this. Instead of tracking all the remote users or subnets that are connected, imagine one route that says, in effect, "this way to the dial portion of the network"!

So route summarization can reduce traffic and computation in an OSPF network. It can render your network more stable. The final benefit is in troubleshooting. A careful design, be it for OSPF or EIGRP, assigns summarizable blocks of addresses to different regions in the network. You can then look at one route saying "this way to Australia", instead of dealing with odds and ends of subnets, looking them up, and then recognizing, "ah, yes, most of Australia seems to be present in the routing table". You probably don't care which links in Australia are up or down, but you do want to be able to route packets to Australia. (The price you pay for summarization is that you may route a packet to a router in Australia before discovering the destination subnet is down. Summarization always has the potential for sub-optimal routes or for extra unnecessary traffic. The trade-off is simplified routing tables versus absolutely optimal behavior.)

## Configuring Route Summarization in OSPF

Before you can configure route summarization in OSPF, you really need a design that allows you to summarize. Let's say you have a need for 10 areas (counting the area 0 backbone). Round up to a power of 2: 16. Let's divide our address space, say 131.108.0.0 /16 into 16 equal sized pieces. There are 256 possible values for the third octet (0 through 255), so that means our pieces will be multiples of 16. The first area (area 0) will be 131.108.0.0 through 131.108.15.255, the second 131.108.16.0 through 131.108.31.255, etc.

Since  $256 - 16 = 240$ , we see that a mask of 255.255.240.0 describes each of the areas -- I've been calling this the "area mask". It describes how much of the address we need to look at to determine what OSPF area the address is from. (See if you can figure out how the 0 bits in this area mask describe that the last 4 bits of octet 3, and all of octet 4, are unimportant in determining the area).

Suppose we've also decided to use the subnet mask 255.255.255.192, a /26. The area mask is a /20, so that gives 6 bits for subnetting within each area. Or 64 subnets per area. Of course, we'd probably use VLSM with mask 255.255.255.252 for WAN links, but that's another topic.

Let's suppose we're configuring an OSPF Area Border Router (ABR), with interface Ethernet 0 in Area 0, and Ethernet 3 in Area 3. We've picked addresses from the right blocks of subnets for each: Ethernet 0 will be 131.108.10.1, and Ethernet 3 will be 131.108.48.1. Then the relevant part of the router configuration would be:

```
interface ethernet 0
ip address 131.108.10.1 255.255.255.192
!
interface ethernet 3
ip address 131.108.48.1 255.255.255.192
!
router ospf 1
network 131.108.0.0 0.0.15.255 area 0
network 131.108.48.0 0.0.15.255 area 3
area 0 range 131.108.0.0 255.255.240.0
area 3 range 131.108.48.0 255.255.240.0
```

The interface configuration is included to emphasize that each interface is given an address from within the appropriate address block for the area the interface is in. The /26 mask shows that we are just doing subnetting as usual. The only thing novel is that we have set ourselves up to use subnets in blocks, with a specific block assigned to each OSPF area.

The network statements are a bit new. The 0.0.15.255 wildcard mask indicates how many addresses, or which interfaces, should be considered to be in area 0. This has nothing to do with forwarding traffic, solely with what interfaces should be running OSPF (those that match one of the area statements), and what area they should claim to be in in the OSPF Hellos (the area specified by the first matching network address/wildcard pattern). Note that neighboring routers will not form an OSPF adjacency if the areas differ, or if any of several other things differ (timers, OSPF authentication string, stubby-ness of the area, etc.).

You can think through the wildcarding scheme, which is the same as for access lists, to see that the above wildcards are correct. (In fact, notice that adding the 15.255 to the start of the address block gets us the last address in the block, 131.108.15.255 or 131.108.63.255).

Another approach is to determine these wildcard masks from the "area mask" stated above. They are the one's complement of it, where the 1 bits are turned to 0 and vice versa. We can do this by subtracting octet-wise from 255.255.255.255. So 255.255.255.255 minus 255.255.240.0 gives ...

0.0.15.255.

That's the nice thing about designing by breaking the address space into equal-sized blocks. It simplifies the computations. We had 16 blocks, hence each covered 16 values in octet 3, hence the area mask was 255.255.240.0, hence the wildcard mask was 0.0.15.255. No binary!

The `area...range` statements are the part of the configuration that tells OSPF to summarize routes. Instead of advertising subnets of 131.108.0.0 in the 131.108.48.0 through 63.255 range, OSPF will just advertise 131.108.48.0 /20 outside the area. We will see it in the routing table as destination 131.108.48.0 with mask 255.255.240.0, exactly what shows in the `area...range` statement. And note that the mask there is the so-called "area mask"!!!

In general, we can summarize routes in OSPF only on Area Border Routers, ABR's. We can either summarize them going out of the original area, or we can summarize them going into another area. Due to the `area...range` statement for area 0, area 0 will show up as 131.108.0.0 /20 within area 3, or 131.108.0.0 with mask 255.255.240.0. Again, note that the area mask was used to configure this, and also appears in the routing table summary route.

## Wrap-Up

I currently plan to wrap up OSPF in next month's article. There's a lot more to OSPF, but we're going to stick to the more basic and essential parts of it in this column. I hope to move on to VLAN's and configuring switches in the next series of articles. Your comments, suggestions, corrections, and topic redirects 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