

Title:

Cisco CCNA / CCNP Certification Exam Lab: Frame Relay Subinterfaces And Split Horizon

Word Count:

751

Summary:

To truly understand Frame Relay subinterfaces and split horizon, you've got to see both in action. Study for your exam and prepare for success with this fully illustrated lab from Chris Bryant, CCIE #12933.

Keywords:

ccna, ccnp, exam, certification, frame, relay, subinterface, split, horizon, rip, network, loopback

Article Body:

Earning your Cisco CCNA and CCNP is a tough proposition, and part of that is the fact that you quickly learn that there's usually more than one way to do things with Cisco routers - and while that's generally a good thing, you better know the ins and outs of all options when it comes to test day and working on production networks. Working with Frame Relay subinterfaces and split horizon is just one such situation.

One reason for the use of subinterfaces is to circumvent the rule of split horizon. You recall from your CCNA studies that split horizon dictates that a route cannot be advertised out the same interface upon which it was learned in the first place. In the following example, R1 is the hub and R2 and R3 are the spokes. All three routers are using their physical interfaces for frame relay connectivity, and they are also running RIPv2 172.12.123.0 /24. Each router is also advertising a loopback interface, using the router number for each octet.

```
R1(config)#int s0
```

```
R1(config-if)#ip address 172.12.123.1 255.255.255.0
```

```
R1(config-if)#no frame inverse
```

```
R1(config-if)#frame map ip 172.12.123.2 122 broadcast
```

```
R1(config-if)#frame map ip 172.12.123.3 123 broadcast
```

```
R1(config-if)#no shut
```

```
R2(config)#int s0
```

```
R2(config-if)#encap frame
```

```
R2(config-if)#no frame inver
```

```
R2(config-if)#frame map ip 172.12.123.1 221 broadcast
```

```
R2(config-if)#frame map ip 172.12.123.3 221 broadcast
```

```
R2(config-if)#ip address 172.12.123.2 255.255.255.0
```

```
R3(config)#int s0
```

```
R3(config-if)#encap frame
```

```
R3(config-if)#no frame inver
```

```
R3(config-if)#frame map ip 172.12.123.1 321 broadcast
```

```
R3(config-if)#frame map ip 172.12.123.2 321 broadcast
```

```
R3(config-if)#ip address 172.12.123.3 255.255.255.0
```

```
R1#show ip route rip
```

```
2.0.0.0/32 is subnetted, 1 subnets
```

```
R          2.2.2.2 [120/1] via 172.12.123.2, 00:00:20, Serial0
```

```
3.0.0.0/32 is subnetted, 1 subnets
```

```
R          3.3.3.3 [120/1] via 172.12.123.3, 00:00:22, Serial0
```

```
R2#show ip route rip
```

```
1.0.0.0/32 is subnetted, 1 subnets
```

```
R          1.1.1.1 [120/1] via 172.12.123.1, 00:00:06, Serial0
```

```
R3#show ip route rip
```

```
1.0.0.0/32 is subnetted, 1 subnets
```

```
R      1.1.1.1 [120/1] via 172.12.123.1, 00:00:04, Serial0
```

The hub router R1 has a route to both loopbacks, but neither spoke has a route to the other spoke's loopback. That's because split horizon prevents R1 from advertising a network via Serial0 if the route was learned on Serial0 to begin with.

We've got two options here, one of which is to disable split horizon on the interface. While doing so will have the desired effect in our little network, disabling split horizon is not a good idea and should be avoided whenever possible. We're not going to do it in this lab, but here is the syntax to do so:

```
R1(config)#interface serial0
```

```
R1(config-if)#no ip split-horizon
```

A better solution is to configure subinterfaces on R1. The IP addressing will have to be revisited, but that's no problem here. R1 and R2 will use 172.12.123.0 /24 to communicate, while R1 and R3 will use 172.12.13.0 /24. R3's serial0 interface will need to be renumbered, so let's look at all three router configurations:

```
R1(config)#interface serial0
```

```
R1(config-if)#encap frame
```

```
R1(config-if)#no frame inverse-arp
```

```
R1(config-if)#no ip address
```

```
R1(config-if)#interface serial0.12 multipoint
```

```
R1(config-subif)#ip address 172.12.123.1 255.255.255.0
```

```
R1(config-subif)#frame map ip 172.12.123.2 122 broadcast
```

```
R1(config-subif)#interface serial0.31 point-to-point
```

```
R1(config-subif)#ip address 172.12.13.1 255.255.255.0
```

```
R1(config-subif)#frame interface-dlci 123
```

```
R2(config)#int s0
```

```
R2(config-if)#ip address 172.12.123.2 255.255.255.0
```

```
R2(config-if)#encap frame
```

```
R2(config-if)#frame map ip 172.12.13.3 221 broadcast
```

```
R2(config-if)#frame map ip 172.12.123.1 221 broadcast
```

```
R3(config)#int s0
```

```
R3(config-if)#ip address 172.12.13.3 255.255.255.0
```

```
R3(config-if)#encap frame
```

```
R3(config-if)#frame map ip 172.12.13.1 321 broadcast
```

```
R3(config-if)#frame map ip 172.12.123.2 321 broadcast
```

A frame map statement always names the REMOTE IP address and the LOCAL DLCI.
Don't forget the broadcast option!

Show frame map shows us that all the static mappings on R1 are up and running.
Note the "static" output, which indicates these mappings are a result of using
the frame map command. Pings are not shown, but all three routers can ping each
other at this point.

```
R1#show frame map
```

```
Serial0 (up): ip 172.12.123.2 dlci 122(0x7A,0x1CA0), static,  
broadcast, CISCO, status defined, active
```

```
Serial0 (up): ip 172.12.13.3 dlci 123(0x7B,0x1CB0), static,  
broadcast, CISCO, status defined, active
```

After the 172.12.13.0 /24 network is added to R1 and R3's RIP configuration, R2 and R3 now have each other's loopback network in their RIP routing tables.

```
R2#show ip route rip
```

```
1.0.0.0/32 is subnetted, 1 subnets
```

```
R      1.1.1.1 [120/1] via 172.12.123.1, 00:00:20, Serial0
```

```
3.0.0.0/32 is subnetted, 1 subnets
```

```
R      3.3.3.3 [120/1] via 172.12.123.1, 00:00:22, Serial0
```

```
R3#show ip route rip
```

```
1.0.0.0/32 is subnetted, 1 subnets
```

```
R      1.1.1.1 [120/1] via 172.12.13.1, 00:00:20, Serial0
```

```
2.0.0.0/32 is subnetted, 1 subnets
```

```
R      2.2.2.2 [120/1] via 172.12.13.1, 00:00:22, Serial0
```

While turning split horizon off is one way to achieve total IP connectivity, doing so can have other unintended results. The use of subinterfaces is a more effective way of allowing the spokes to see the hub's loopback network.