



IEWB-RS Technology Labs

Frame Relay

Brian Dennis, CCIE # 2210 (R&S / ISP Dial / Security / Service Provider)
Brian McGahan, CCIE # 8583 (R&S / Service Provider)

Copyright Information

Copyright © 2003 - 2007 Internetwork Expert, Inc. All rights reserved.

The following publication, ***CCIE Routing and Switching Lab Workbook***, was developed by Internetwork Expert, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means without the prior written permission of Internetwork Expert, Inc.

Cisco®, Cisco® Systems, CCIE, and Cisco Certified Internetwork Expert, are registered trademarks of Cisco® Systems, Inc. and/or its affiliates in the U.S. and certain countries.

All other products and company names are the trademarks, registered trademarks, and service marks of the respective owners. Throughout this manual, Internetwork Expert, Inc. has used its best efforts to distinguish proprietary trademarks from descriptive names by following the capitalization styles used by the manufacturer.

Disclaimer

The following publication, ***CCIE Routing and Switching Lab Workbook***, is designed to assist candidates in the preparation for Cisco Systems' CCIE Routing & Switching Lab exam. While every effort has been made to ensure that all material is as complete and accurate as possible, the enclosed material is presented on an "as is" basis. Neither the authors nor Internetwork Expert, Inc. assume any liability or responsibility to any person or entity with respect to loss or damages incurred from the information contained in this workbook.

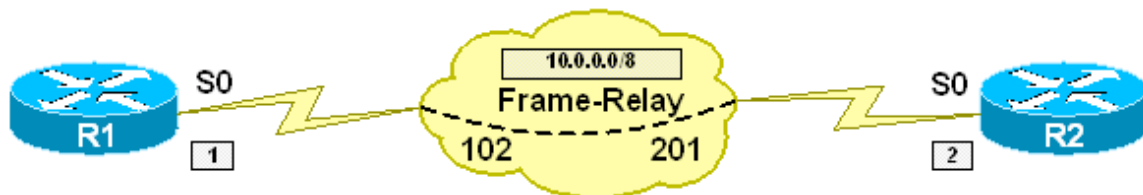
This workbook was developed by Internetwork Expert, Inc. and is an original work of the aforementioned authors. Any similarities between material presented in this workbook and actual CCIE™ lab material is completely coincidental.

FRAME-RELAY	1
FRAME-RELAY STATIC MAPPING 1	4
FRAME-RELAY INVERSE-ARP & STATIC MAPPINGS.....	7
FRAME-RELAY MULTIPOINT INTERFACES: INVERSE-ARP	10
FRAME-RELAY MULTIPOINT INTERFACES: STATIC MAPPINGS	13
FRAME-RELAY MULTIPOINT INTERFACES: INVERSE-ARP & STATIC MAPPINGS	16
FRAME-RELAY POINT-TO-POINT SUBINTERFACES	19
FRAME-RELAY POINT-TO-POINT AND MAIN INTERFACES: INVERSE-ARP	22
FRAME-RELAY POINT-TO-POINT AND MAIN INTERFACES: STATIC MAPPINGS	25
FRAME-RELAY POINT-TO-POINT AND MULTIPOINT SUBINTERFACES: INVERSE-ARP	28
FRAME-RELAY POINT-TO-POINT AND MULTIPOINT SUBINTERFACES: STATIC MAPPING	31
FRAME-RELAY MAIN INTERFACE: INVERSE-ARP, MULTIPOINT INTERFACE: INVERSE-ARP	34
FRAME-RELAY MAIN INTERFACE: INVERSE-ARP, MULTIPOINT INTERFACE: STATIC MAPPING	37
FRAME-RELAY MAIN INTERFACE: STATIC MAPPING, MULTIPOINT INTERFACE: INVERSE-ARP	40
FRAME-RELAY HUB-AND-SPOKE, MAIN INTERFACES W/ INVERSE-ARP & STATIC MAPPINGS..	50
FRAME-RELAY HUB-AND-SPOKE, MAIN INTERFACES W/ STATIC MAPPINGS	54
FRAME-RELAY HUB-AND-SPOKE, MAIN INTERFACES W/ INVERSE-ARP & POINT-TO-POINT SUBINTERFACES	58
FRAME-RELAY HUB-AND-SPOKE, MAIN INTERFACES W/ STATIC MAPPINGS & POINT-TO-POINT SUBINTERFACES	62

Frame-Relay

Frame-Relay Inverse-ARP

Objective: Configure a Frame Relay PVC connection using Frame Relay Inverse-ARP



Directions

- Configure R1's main serial interface with the IP address 10.0.0.1/8
- Configure R2's main serial interface with the IP address 10.0.0.2/8
- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Verify dynamic resolution via Frame Relay Inverse-ARP

Ask Yourself

- Is the main interface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the different methods to configure this resolution?
- What configuration steps are required to enable Frame Relay Inverse-ARP?

Final Configuration

```

R1:
interface Serial0/0
 ip address 10.0.0.1 255.0.0.0
 encapsulation frame-relay

R2:
interface Serial0/0
 ip address 10.0.0.2 255.0.0.0
 encapsulation frame-relay

```

Verification

```

R1#ping 10.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:

```

y

```
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms

R1#ping 255.255.255.255 repeat 1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.2, 8 ms

R2#ping 255.255.255.255 repeat 1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 12 ms

R1#show frame-relay pvc 102

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 12          output pkts 12          in bytes 1108
  out bytes 1108        dropped pkts 0          in pkts dropped 0
  out pkts dropped 0    out bytes dropped 0
  in FECN pkts 0        in BECN pkts 0        out FECN pkts 0
  out BECN pkts 0        in DE pkts 0          out DE pkts 0
  out bcast pkts 2      out bcast bytes 68
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:05:50, last time pvc status changed 00:02:34

R1#show frame-relay map
Serial0/0 (up): ip 10.0.0.2 dlci 102(0x66,0x1860), dynamic,
                broadcast,, status defined, active

R2#show frame-relay pvc 201

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 13          output pkts 12          in bytes 1142
  out bytes 1108        dropped pkts 0          in pkts dropped 0
  out pkts dropped 0    out bytes dropped 0
  in FECN pkts 0        in BECN pkts 0        out FECN pkts 0
  out BECN pkts 0        in DE pkts 0          out DE pkts 0
  out bcast pkts 2      out bcast bytes 68
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:05:19, last time pvc status changed 00:02:39
R2#

R2#show frame-relay map
Serial0/0 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), dynamic,
                broadcast,, status defined, active
```

Breakdown

NBMA configurations fundamentally come in two different forms, multipoint and point-to-point. Multipoint configurations allow you to connect to multiple devices on one segment, while point-to-point configurations can only connect to one other endpoint. With multipoint configurations such as the main interface in Frame Relay or ATM, layer 3 to layer 2 protocol resolution is required so that the router knows what layer 2 address to put in the packet when it is encapsulated.

Layer 3 to layer 2 mapping can occur dynamically through a protocol such as Inverse-ARP, or statically through statements such as the frame-relay map or dialer map in Frame Relay and ISDN respectively. In the above case, resolution is obtained automatically through Frame Relay Inverse-ARP. Note that there are no specific configuration steps required to enable Frame Relay Inverse-ARP on the interface other than enabling Frame Relay encapsulation and configuring a protocol on the interface. Once the interface is enabled (no shutdown), the router will send Frame Relay Inverse-ARP requests out all DLCIs learned via LMI for all configured protocols on the interface (i.e. IP, IPX, IPv6 etc.).

To verify that Frame Relay Inverse-ARP was successful, issue the `show frame-relay map` command. The keyword `dynamic` indicates the mapping occurred due to Frame Relay Inverse-ARP.

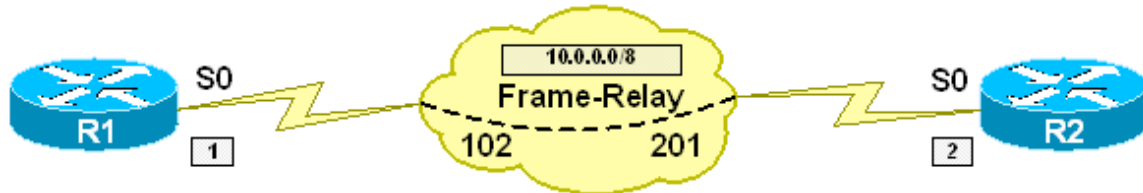


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Static Mapping 1

Objective: Configure a Frame Relay PVC connection using static frame-relay map statements



Directions

- Configure R1's main serial interface with the IP address 10.0.0.1/8
- Configure R2's main serial interface with the IP address 10.0.0.2/8
- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Configure R1 with a static layer 3 to layer 2 mapping to use DLCI 102 to reach the IP address 10.0.0.2
- Configure R2 with a static layer 3 to layer 2 mapping to use DLCI 201 to reach the IP address 10.0.0.1
- Ensure that broadcast and multicast traffic can be sent across the circuit from both ends of the link

Ask Yourself

- Is the main interface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the different methods to configure this resolution?
- What steps are required to configure a static layer 3 to layer 2 resolution?

Final Configuration

```
R1:
interface Serial0/0
 ip address 10.0.0.1 255.0.0.0
 encapsulation frame-relay
 frame-relay map ip 10.0.0.2 102 broadcast
```

```
R2:
interface Serial0/0
 ip address 10.0.0.2 255.0.0.0
 encapsulation frame-relay
 frame-relay map ip 10.0.0.1 201 broadcast
```

Verification

```
R1#ping 10.0.0.2
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms
```

```
R1#ping 255.255.255.255 repeat 1
```

```
Type escape sequence to abort.  
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:  
  
Reply to request 0 from 10.0.0.2, 8 ms
```

```
R2#ping 255.255.255.255 repeat 1
```

```
Type escape sequence to abort.  
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:  
  
Reply to request 0 from 10.0.0.1, 12 ms
```

```
R1#show frame-relay map  
Serial0/0 (up): ip 10.0.0.2 dlci 102(0x66,0x1860), static,  
                broadcast,  
                CISCO, status defined, active
```

```
R2#show frame-relay map  
Serial0/0 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), static,  
                broadcast,  
                CISCO, status defined, active
```

Breakdown

NBMA configurations fundamentally come in two different forms, multipoint and point-to-point. Multipoint configurations allow you to connect to multiple devices on one segment, while point-to-point configurations can only connect to one other endpoint. With multipoint configurations such as the main interface in Frame Relay or ATM, layer 3 to layer 2 protocol resolution is required so that the router knows what layer 2 address to put in the packet when it is encapsulated.

Layer 3 to layer 2 mapping can occur dynamically through a protocol such as Inverse-ARP, or statically through statements such as the frame-relay map or dialer map in Frame Relay and ISDN respectively. In the above case, resolution is obtained manually through the frame-relay map statement. The `frame-relay map` command manually associates a remote layer 3 address, like an ip address, to a local layer 2 address, or DLCI.

In the above configuration, the command `frame-relay map ip 10.0.0.2 102 broadcast` is configured on R1. This tells R1 that if it needs to send a packet towards 10.0.0.2, use the DLCI number 102 in the frame-relay header when the packet is encapsulated on the line. In addition to this, if a broadcast or multicast

packet is sent to interface Serial0 for transmission, it will be replicated out circuit 102. This is verified by sending ICMP echo traffic to the address 255.255.255.255 (all subnet broadcast) from both R1 and R2, and receiving a reply from the other end of the link. The implication of broadcast replication on a non-broadcast circuit will be covered in later labs.

To verify the static Frame Relay mapping, issue the `show frame-relay map` command. Note that unlike the previous mapping created through Frame Relay Inverse-ARP, the output with the `frame-relay map` shows the keyword `static`.

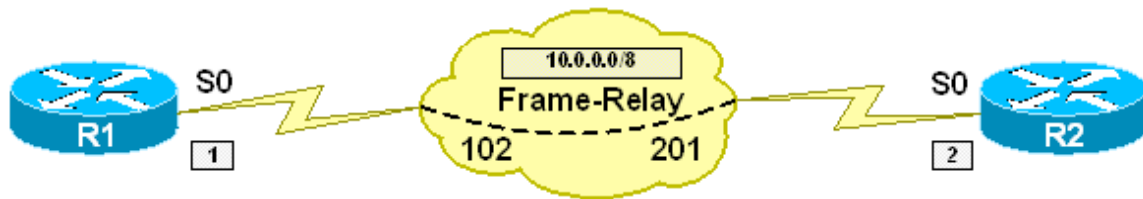


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Inverse-ARP & Static Mappings

Objective: Configure a Frame Relay PVC connection between R1 and R2 using Frame Relay Inverse-ARP on R1 and a static frame-relay map statement on R2



Directions

- Configure R1's main serial interface with the IP address 10.0.0.1/8
- Configure R2's main serial interface with the IP address 10.0.0.2/8
- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Configure R2 with a static layer 3 to layer 2 mapping to use DLCI 201 to reach the IP address 10.0.0.1
- Ensure that broadcast and multicast traffic can be sent across the circuit from both ends of the link

Ask Yourself

- Is the main interface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the different methods to configure this resolution?
- What steps are required to configure dynamic layer 3 to layer 2 resolution via Frame Relay Inverse-ARP?
- What steps are required to configure a static layer 3 to layer 2 resolution?

Final Configuration

```
R1:
interface Serial0/0
 ip address 10.0.0.1 255.0.0.0
 encapsulation frame-relay

R2:
interface Serial0/0
 ip address 10.0.0.2 255.0.0.0
 encapsulation frame-relay
 frame-relay map ip 10.0.0.1 201 broadcast
```

Verification

```
R1#ping 10.0.0.2
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

```
R1#ping 255.255.255.255 repeat 1
```

```
Type escape sequence to abort.  
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:  
  
Reply to request 0 from 10.0.0.2, 8 ms
```

```
R2#ping 255.255.255.255 repeat 1
```

```
Type escape sequence to abort.  
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:  
  
Reply to request 0 from 10.0.0.1, 12 ms
```

```
R1#show frame-relay map  
Serial0/0 (up): ip 10.0.0.2 dlci 102(0x66,0x1860), dynamic,  
                broadcast, status defined, active
```

```
R2#show frame-relay map  
Serial0/0 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), static,  
                broadcast,  
                CISCO, status defined, active
```

Breakdown

NBMA configurations fundamentally come in two different forms, multipoint and point-to-point. Multipoint configurations allow you to connect to multiple devices on one segment, while point-to-point configurations can only connect to one other endpoint. With multipoint configurations such as the main interface in Frame Relay or ATM, layer 3 to layer 2 protocol resolution is required so that the router knows what layer 2 address to put in the packet when it is encapsulated.

Layer 3 to layer 2 mapping can occur dynamically through a protocol such as Inverse-ARP, or statically through statements such as the frame-relay map or dialer map in Frame Relay and ISDN respectively. In the above case, resolution is obtained automatically on R1 through Frame Relay Inverse-ARP, and manually through the `frame-relay map` statement on R2.

The above configuration dictates that layer 3 to layer 2 resolution is only locally significant, since a dynamic mapping is used on one side, while a static mapping is used on the other side. Like in the previous cases, this can be verified by issuing the `show frame-relay map` command in privilege level mode.

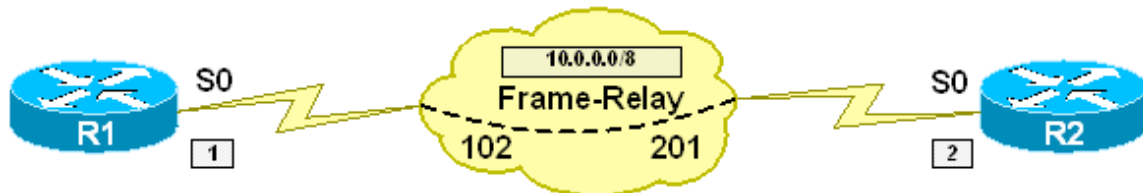


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Multipoint Interfaces: Inverse-ARP

Objective: Configure a Frame Relay PVC connection between R1 and R2 using multipoint subinterfaces and Frame Relay Inverse-ARP



Directions

- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Create a multipoint subinterface S0.1 on R1
- Create a multipoint subinterface S0.1 on R2
- Configure the subinterface on R1 with the IP address 10.0.0.1/8
- Configure the subinterface on R2 with the IP address 10.0.0.2/8
- Assign the DLCI 102 to the subinterface on R1
- Assign the DLCI 201 to the subinterface on R1

Ask Yourself

- Is a multipoint subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the different methods to configure this resolution?
- What steps are required to configure dynamic layer 3 to layer 2 resolution via Frame Relay Inverse-ARP?
- When DLCIs are learned via LMI from the Frame Relay cloud, where are they assigned?
- How can I change where they are assigned?

Final Configuration

```
R1:
interface Serial0/0
  encapsulation frame-relay
!
interface Serial0/0.1 multipoint
  ip address 10.0.0.1 255.0.0.0
  frame-relay interface-dlci 102
```

```
R2:
interface Serial0/0
  encapsulation frame-relay
```

y

```
!
interface Serial0/0.1 multipoint
 ip address 10.0.0.2 255.0.0.0
 frame-relay interface-dlci 201
```

Verification

R1#ping 10.0.0.2

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms
```

R1#show frame-relay pvc 102

```
PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

  input pkts 2           output pkts 1           in bytes 68
  out bytes 34          dropped pkts 0          in pkts dropped 0
  out pkts dropped 0    out bytes dropped 0
  in FECN pkts 0        in BECN pkts 0          out FECN pkts 0
  out BECN pkts 0        in DE pkts 0            out DE pkts 0
  out bcast pkts 1      out bcast bytes 34
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:07:18, last time pvc status changed 00:06:01
```

R1#show frame-relay map

```
Serial0/0.1 (up): ip 10.0.0.2 dlci 102(0x66,0x1860), dynamic,
                  broadcast,, status defined, active
```

R2#show frame-relay pvc 201

```
PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

  input pkts 6           output pkts 6           in bytes 554
  out bytes 554         dropped pkts 0          in pkts dropped 0
  out pkts dropped 0    out bytes dropped 0
  in FECN pkts 0        in BECN pkts 0          out FECN pkts 0
  out BECN pkts 0        in DE pkts 0            out DE pkts 0
  out bcast pkts 1      out bcast bytes 34
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:08:18, last time pvc status changed 00:07:12
```

R2#show frame-relay map

```
Serial0/0.1 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), dynamic,
                  broadcast,, status defined, active
```

Breakdown

Like the main interface in Frame Relay, multipoint subinterfaces require layer 3 to layer 2 resolution for all protocols configured on the interface. This resolution can occur dynamically through Frame Relay Inverse-ARP, or statically through the `frame-relay map` command.

To create a multipoint subinterface, first encapsulate frame-relay on the main interface with the `encapsulation frame-relay` command. Next, create the subinterface by issuing the `interface serial0.1 multipoint` command. Once the subinterface is created, configure the layer 3 protocols, such as IP, IPX, etc., and assign the layer 2 circuit. By default, all DLCIs learned from the Frame Relay cloud via LMI are assigned to the main interface. Therefore, the DLCIs used in the above configuration must be moved from the main interface to the multipoint subinterface. This can be accomplished either with the `frame-relay interface-dlci` command or the `frame-relay map` command.

The difference between the two statements is that the `frame-relay interface-dlci` command does not enable layer 3 to layer 2 resolution. Instead, it simply assigns the circuit to the interface. Once the circuit is assigned, Frame Relay Inverse-ARP requests will be sent out the circuit or circuits that have been assigned to the interface.

To verify that the DLCIs have been assigned to the subinterface, issue the `show frame-relay pvc` command. Like in the previous cases, layer 3 to layer 2 protocol resolution is verified by issuing the `show frame-relay map` command in privilege level mode.

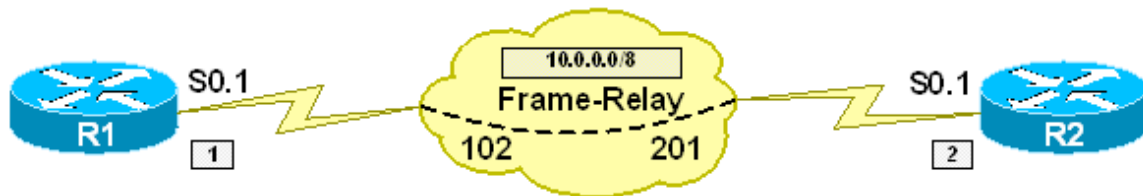


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Multipoint Interfaces: Static Mappings

Objective: Configure a Frame Relay PVC connection between R1 and R2 using multipoint subinterfaces with static layer 3 to layer 2 protocol mappings



Directions

- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Create a multipoint subinterface S0.1 on R1
- Create a multipoint subinterface S0.1 on R2
- Configure the subinterface on R1 with the IP address 10.0.0.1/8
- Configure the subinterface on R2 with the IP address 10.0.0.2/8
- Configure a static mapping on R1 to use DLCI 102 to reach the IP address 10.0.0.2
- Configure a static mapping on R2 to use DLCI 201 to reach the IP address 10.0.0.1
- Ensure that broadcast and multicast support is enabled on the circuit

Ask Yourself

- Is a multipoint subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the different methods to configure this resolution?
- When DLCIs are learned via LMI from the Frame Relay cloud, where are they assigned?
- How can I change where they are assigned?
- What steps are required to configure static layer 3 to layer 2 resolution?
- Does configuring this resolution automatically change what interface the DLCI is assigned to?

Final Configuration

```
R1:
interface Serial0/0
  no ip address
  encapsulation frame-relay
!
```

y


```
interface Serial0/0.1 multipoint
 ip address 10.0.0.1 255.0.0.0
 frame-relay map ip 10.0.0.2 102 broadcast
```

R2:

```
interface Serial0/0
 encapsulation frame-relay
 !
interface Serial0/0.1 multipoint
 ip address 10.0.0.2 255.0.0.0
 frame-relay map ip 10.0.0.1 201 broadcast
```

Verification

R1#ping 10.0.0.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms

R1#ping 255.255.255.255 repeat 1

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.2, 8 ms

R2#ping 255.255.255.255 repeat 1

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 12 ms

R1#show frame-relay pvc 102

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

```
input pkts 2          output pkts 1          in bytes 68
out bytes 34          dropped pkts 0         in pkts dropped 0
out pkts dropped 0    out bytes dropped 0
in FECN pkts 0       in BECN pkts 0        out FECN pkts 0
out BECN pkts 0       in DE pkts 0          out DE pkts 0
out bcast pkts 1     out bcast bytes 34
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:07:18, last time pvc status changed 00:06:01
```

R1#show frame-relay map

```
Serial0/0.1 (up): ip 10.0.0.2 dlci 102(0x66,0x1860), static,
                broadcast,
                CISCO, status defined, active
```

R2#show frame-relay pvc 201

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

```
DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

input pkts 6          output pkts 6          in bytes 554
out bytes 554        dropped pkts 0        in pkts dropped 0
out pkts dropped 0    out bytes dropped 0
in FECN pkts 0       in BECN pkts 0       out FECN pkts 0
out BECN pkts 0      in DE pkts 0         out DE pkts 0
out bcast pkts 1     out bcast bytes 34
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:08:18, last time pvc status changed 00:07:12

R2#show frame-relay map
Serial0/0.1 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), static,
                broadcast,
                CISCO, status defined, active
```

Breakdown

Like the main interface in Frame Relay, multipoint subinterfaces require layer 3 to layer 2 resolution for all protocols configured on the interface. This resolution can occur dynamically through Frame Relay Inverse-ARP, or statically through the `frame-relay map` command.

R1 and R2 are configured with static layer 3 to layer 2 resolution to resolve each other's IP addresses to the local Frame Relay DLCIs. This is accomplished with the `frame-relay map` command on both subinterfaces. Note that the command `frame-relay interface-dlci` is not required on either of these devices, because issuing the `frame-relay map` statement also assigns the circuit to the interface.

To verify that the DLCIs have been assigned to the subinterface, issue the `show frame-relay pvc` command. Like in the previous cases, layer 3 to layer 2 protocol resolution is verified by issuing the `show frame-relay map` command in privilege level mode.

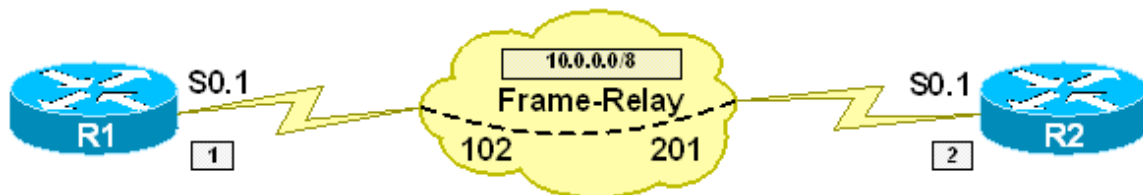


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Multipoint Interfaces: Inverse-ARP & Static Mappings

Objective: Configure a Frame Relay PVC connection between R1 and R2 using multipoint subinterfaces with Frame Relay Inverse-ARP on R1 and static mappings on R2



Directions

- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Create a multipoint subinterface S0.1 on R1
- Create a multipoint subinterface S0.1 on R2
- Configure the subinterface on R1 with the IP address 10.0.0.1/8
- Configure the subinterface on R2 with the IP address 10.0.0.2/8
- Assign the DLCI 102 to the subinterface on R1
- Configure a static mapping on R2 to use DLCI 201 to reach the IP address 10.0.0.1
- Ensure that broadcast and multicast support is enabled on the circuit

Ask Yourself

- Is a multipoint subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the different methods to configure this resolution?
- What steps are required to configure dynamic layer 3 to layer 2 resolution via Frame Relay Inverse-ARP?
- When DLCIs are learned via LMI from the Frame Relay cloud, where are they assigned?
- How can I change where they are assigned to?
- What steps are required to configure static layer 3 to layer 2 resolution?
- Does configuring this resolution automatically change what interface the DLCI is assigned to?

Final Configuration

```
R1:
interface Serial0/0
 encapsulation frame-relay
```

y

```

!
interface Serial0/0.1 multipoint
 ip address 10.0.0.1 255.0.0.0
 frame-relay interface-dlci 102

R2:
interface Serial0/0
 encapsulation frame-relay
!
interface Serial0/0.1 multipoint
 ip address 10.0.0.2 255.0.0.0
 frame-relay map ip 10.0.0.1 201 broadcast

```

Verification

R1#ping 10.0.0.2

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms

```

R1#ping 255.255.255.255 repeat 1

```

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.2, 8 ms

```

R2#ping 255.255.255.255 repeat 1

```

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 12 ms

```

R1#show frame-relay pvc 102

```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

  input pkts 2           output pkts 1           in bytes 68
  out bytes 34          dropped pkts 0          in pkts dropped 0
  out pkts dropped 0     out bytes dropped 0
  in FECN pkts 0        in BECN pkts 0         out FECN pkts 0
  out BECN pkts 0        in DE pkts 0           out DE pkts 0
  out bcast pkts 1      out bcast bytes 34
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:07:18, last time pvc status changed 00:06:01

```

R1#show frame-relay map

```

Serial0/0.1 (up): ip 10.0.0.2 dlci 102(0x66,0x1860), dynamic,
                broadcast,, status defined, active

```

R2#show frame-relay pvc 201

```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

```

```
DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

input pkts 6          output pkts 6          in bytes 554
out bytes 554        dropped pkts 0        in pkts dropped 0
out pkts dropped 0    out bytes dropped 0
in FECN pkts 0      in BECN pkts 0      out FECN pkts 0
out BECN pkts 0     in DE pkts 0        out DE pkts 0
out bcast pkts 1     out bcast bytes 34
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:08:18, last time pvc status changed 00:07:12

R2#show frame-relay map
Serial0/0.1 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), static,
                broadcast,
                CISCO, status defined, active
```

Breakdown

Like the main interface in Frame Relay, multipoint subinterfaces require layer 3 to layer 2 resolution for all protocols configured on the interface. This resolution can occur dynamically through Frame Relay Inverse-ARP, or statically through the `frame-relay map` command.

Like the previous example, R1 is using Frame Relay Inverse-ARP on circuit 102 to resolve the IP address of R2. This is accomplished by assigning an IP address to its subinterface Serial0.1, and applying the layer 2 circuit to the interface with the `frame-relay interface-dlci 102` command.

R2 is configured with static layer 3 to layer 2 resolution to resolve R1's IP address 10.0.0.1 to the local Frame Relay DLCI 201. This is accomplished with the `frame-relay map ip 10.0.0.1 201 broadcast` command on R2's subinterface. Note that the command `frame-relay interface-dlci` is not required on R2, because issuing the `frame-relay map` statement also assigns the circuit to the interface.

This exercise dictates that layer 3 to layer 2 resolution is only locally significant, as R1 is doing dynamic resolution while R2 is doing static resolution. To verify that the DLCIs have been assigned to the subinterface, issue the `show frame-relay pvc` command. Like in the previous cases, layer 3 to layer 2 protocol resolution is verified by issuing the `show frame-relay map` command in privilege level mode.

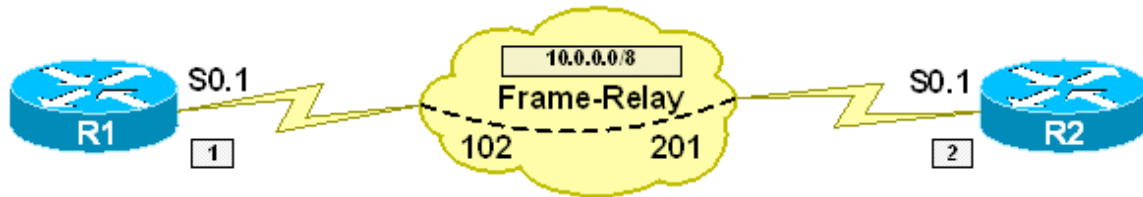


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Point-to-Point Subinterfaces

Objective: Configure a Frame Relay PVC connection between R1 and R2 using point-to-point subinterfaces



Directions

- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Create a point-to-point subinterface S0.1 on R1 and R2.
- Configure the subinterface on R1 with the IP address 10.0.0.1/8
- Configure the subinterface on R2 with the IP address 10.0.0.2/8
- Assign the DLCI 102 to the subinterface on R1
- Assign the DLCI 201 to the subinterface on R2

Ask Yourself

- Is a point-to-point subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If not, what other configuration steps are required?
- When DLCIs are learned via LMI from the Frame Relay cloud, where are they assigned?
- How can I change where they are assigned?

Final Configuration

```

R1:
interface Serial0/0
  encapsulation frame-relay
  !
interface Serial0/0.1 point-to-point
  ip address 10.0.0.1 255.0.0.0
  frame-relay interface-dlci 102

R2:
interface Serial0/0
  encapsulation frame-relay
  !
interface Serial0/0.2 point-to-point
  ip address 10.0.0.2 255.0.0.0
  frame-relay interface-dlci 201

```

Verification

R1#ping 10.0.0.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

R1#ping 255.255.255.255 repeat 1

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.2, 48 ms

R2#ping 255.255.255.255 repeat 1

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 8 ms

R1#show frame-relay pvc 102

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

input pkts 12	output pkts 24	in bytes 1248
out bytes 5340	dropped pkts 0	in pkts dropped 0
out pkts dropped 0	out bytes dropped 0	
in FECN pkts 0	in BECN pkts 0	out FECN pkts 0
out BECN pkts 0	in DE pkts 0	out DE pkts 0
out bcast pkts 13	out bcast bytes 4196	
5 minute input rate 0 bits/sec, 0 packets/sec		
5 minute output rate 0 bits/sec, 0 packets/sec		
pvc create time 00:09:11, last time pvc status changed 00:08:33		

R1#show frame-relay map

Serial0/0.1 (up): point-to-point dlci, dlci 102(0x66,0x1860), broadcast status defined, active

R2#show frame-relay pvc 201

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

input pkts 25	output pkts 12	in bytes 5681
out bytes 1248	dropped pkts 0	in pkts dropped 0
out pkts dropped 0	out bytes dropped 0	
in FECN pkts 0	in BECN pkts 0	out FECN pkts 0
out BECN pkts 0	in DE pkts 0	out DE pkts 0
out bcast pkts 1	out bcast bytes 104	
5 minute input rate 0 bits/sec, 0 packets/sec		
5 minute output rate 0 bits/sec, 0 packets/sec		
pvc create time 00:10:46, last time pvc status changed 00:10:46		

R2#show frame-relay map

y

```
Serial0/0.1 (up): point-to-point dlci, dlci 201(0xC9,0x3090), broadcast
                  status defined, active
```

Breakdown

Unlike multipoint interfaces, point-to-point non-broadcast interfaces do not require layer 3 to layer 2 protocol resolution, as there is only one possible layer 2 address to use when a packet is encapsulated on the line. Therefore, the only required step in configuring a point-to-point non-broadcast circuit is to assign the layer 2 circuit to the interface.

In the above example, point-to-point interfaces are defined on R1 and R2. Next, the Frame Relay DLCIs are assigned to the subinterfaces by issuing the `frame-relay interface-dlci` command, followed by the appropriate PVC numbers.

When the configuration is verified with the `show frame-relay map` command, a unicast layer 3 address is not associated with the DLCI. Instead, the output `point-to-point dlci` indicates that any traffic transiting the subinterface will use the DLCI specified on the interface. In addition to this, broadcast support is automatically enabled on point-to-point NBMA circuits. This can be seen by the output of the `show frame-relay map` command, as the `broadcast` keyword is associated with the mapping.

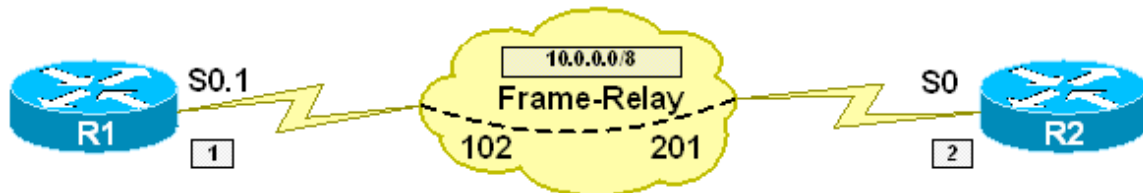


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Point-to-Point Subinterfaces and Main Interfaces: Inverse-ARP

Objective: Configure a Frame Relay PVC connection between R1 and R2 using a point-to-point subinterface on R1 and the main interface on R2 using Frame Relay Inverse-ARP

**Directions**

- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Create a point-to-point subinterface S0.1 on R1
- Configure the subinterface on R1 with the IP address 10.0.0.1/8
- Configure the main interface on R2 with the IP address 10.0.0.2/8
- Assign the DLCI 102 to the subinterface on R1

Ask Yourself

- Is a point-to-point subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If not, what other configuration steps are required?
- When DLCIs are learned via LMI from the Frame Relay cloud, where are they assigned?
- How can I change where they are assigned to?
- Is the main interface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the methods to configure this resolution?
- What resolution method will be enabled by default?

Final Configuration

```

R1:
interface Serial0/0
 encapsulation frame-relay
 !
interface Serial0/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 frame-relay interface-dlci 102

R2:
interface Serial0/0
 ip address 10.0.0.2 255.0.0.0
 encapsulation frame-relay

```

Verification

R1#ping 10.0.0.2

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

```

R1#ping 255.255.255.255 repeat 1

```

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.2, 48 ms

```

R2#ping 255.255.255.255 repeat 1

```

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 8 ms

```

R1#show frame-relay pvc 102

```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

  input pkts 12          output pkts 24          in bytes 1248
  out bytes 5340        dropped pkts 0          in pkts dropped 0
  out pkts dropped 0    out bytes dropped 0
  in FECN pkts 0        in BECN pkts 0         out FECN pkts 0
  out BECN pkts 0       in DE pkts 0           out DE pkts 0
  out bcast pkts 13     out bcast bytes 4196
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:09:11, last time pvc status changed 00:08:33

```

R1#show frame-relay map

```

Serial0/0.1 (up): point-to-point dlci, dlci 102(0x66,0x1860), broadcast
                  status defined, active

```

```
R2#show frame-relay pvc 201
PVC Statistics for interface Serial0/0 (Frame Relay DTE)
DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 24          output pkts 13          in bytes 4085
  out bytes 1212        dropped pkts 0          in pkts dropped 0
  out pkts dropped 0    out bytes dropped 0
  in FECN pkts 0        in BECN pkts 0         out FECN pkts 0
  out BECN pkts 0       in DE pkts 0           out DE pkts 0
  out bcast pkts 2      out bcast bytes 68
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:04:06, last time pvc status changed 00:03:57

R2#show frame-relay map
Serial0/0 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), dynamic,
                broadcast,, status defined, active
```

Breakdown

In the above example, a point-to-point interface is used on R1, while the main interface is used on R2. On R1, layer 3 to layer 2 protocol resolution is not required. Instead, the circuit is simply assigned to the point-to-point subinterface with the `frame-relay interface-dlci` command. On R2, the circuit 102 is assigned to the main interface by default when it is learned via LMI. Since IP is configured on the main interface, a Frame Relay Inverse-ARP request is sent out the circuit for IP, and R1 replies with its IP address 10.0.0.1.

When the configuration is verified with the `show frame-relay map` command, a point-to-point resolution is seen on R1 with broadcast support, while a dynamic unicast layer 3 to layer 2 mapping is seen on R2, also with broadcast support. This configuration dictates that not only is layer 3 to layer 2 resolution only locally significant, the type of interface, whether a main interface, multipoint subinterface, or point-to-point subinterface, is also only locally significant.

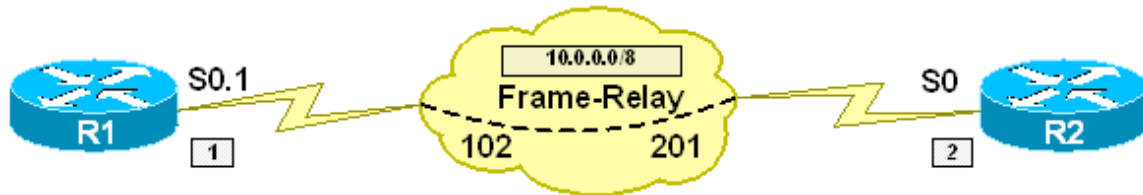


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Point-to-Point Subinterfaces and Main Interfaces: Static Mappings

Objective: Configure a Frame Relay PVC connection between R1 and R2 using a point-to-point subinterface on R1 and the main interface on R2 using a static layer 3 to layer 2 resolution



Directions

- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Create a point-to-point subinterface S0.1 on R1
- Configure the subinterface on R1 with the IP address 10.0.0.1/8
- Configure the main interface on R2 with the IP address 10.0.0.2/8
- Assign the DLCI 102 to the subinterface on R1
- Configure a static layer 3 to layer 2 resolution on R2 to use DLCI 201 to reach the IP address 10.0.0.1
- Ensure that broadcast and multicast support is enabled on the circuit

Ask Yourself

- Is a point-to-point subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If not, what other configuration steps are required?
- When DLCIs are learned via LMI from the Frame Relay cloud, where are they assigned?
- How can I change where they are assigned?
- Is the main interface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the methods to configure this resolution?

Final Configuration

```
R1:
interface Serial0/0
  encapsulation frame-relay
!
interface Serial0/0.1 point-to-point
ip address 10.0.0.1 255.0.0.0
```

```
frame-relay interface-dlci 102
```

R2:

```
interface Serial0/0
ip address 10.0.0.2 255.0.0.0
encapsulation frame-relay
frame-relay map ip 10.0.0.1 201 broadcast
```

Verification

```
R1#ping 10.0.0.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

```
R1#ping 255.255.255.255 repeat 1
```

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.2, 48 ms

```
R2#ping 255.255.255.255 repeat 1
```

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 8 ms

```
R1#show frame-relay pvc 102
```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

input pkts 12	output pkts 24	in bytes 1248
out bytes 5340	dropped pkts 0	in pkts dropped 0
out pkts dropped 0	out bytes dropped 0	
in FECN pkts 0	in BECN pkts 0	out FECN pkts 0
out BECN pkts 0	in DE pkts 0	out DE pkts 0
out bcast pkts 13	out bcast bytes 4196	

5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

pvc create time 00:09:11, last time pvc status changed 00:08:33

```
R1#show frame-relay map
```

Serial0/0.1 (up): point-to-point dlci, dlci 102(0x66,0x1860), broadcast status defined, active

```
R2#show frame-relay pvc 201
```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

input pkts 24	output pkts 13	in bytes 4085
out bytes 1212	dropped pkts 0	in pkts dropped 0

y

```
out pkts dropped 0          out bytes dropped 0
in FECN pkts 0            in BECN pkts 0          out FECN pkts 0
out BECN pkts 0          in DE pkts 0            out DE pkts 0
out bcast pkts 2        out bcast bytes 68
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:04:06, last time pvc status changed 00:03:57

R2#show frame-relay map
Serial0/0 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), static,
                broadcast,
                CISCO, status defined, active
```

Breakdown

In the above example, a point-to-point interface is used on R1, while the main interface is used on R2. On R1, layer 3 to layer 2 protocol resolution is not required. Instead, the circuit is simply assigned to the point-to-point subinterface with the `frame-relay interface-dlci` command. On R2, a static layer 3 to layer 2 resolution is configured with the `frame-relay map ip 10.0.0.1 201 broadcast` statement.

When the configuration is verified with the `show frame-relay map` command, a point-to-point resolution is seen on R1 with broadcast support, while a static unicast layer 3 to layer 2 mapping is seen on R2, also with broadcast support. This configuration dictates that not only is layer 3 to layer 2 resolution only locally significant, the type of interface, whether a main interface, multipoint subinterface, or point-to-point subinterface, is also only locally significant.

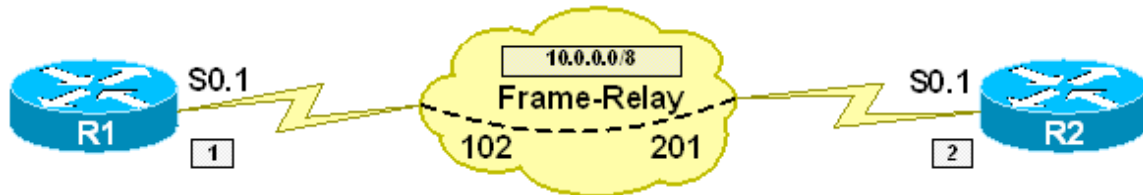


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Point-to-Point Subinterfaces and Multipoint Subinterfaces: Inverse-ARP

Objective: Configure a Frame Relay PVC connection between R1 and R2 using a point-to-point subinterface on R1 and a multipoint subinterface on R2 using Frame Relay Inverse-ARP



Directions

- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Create a point-to-point subinterface S0.1 on R1
- Create a multipoint subinterface S0.1 on R2
- Configure the subinterface on R1 with the IP address 10.0.0.1/8
- Configure the subinterface on R2 with the IP address 10.0.0.2/8
- Assign the DLCI 102 to the subinterface on R1
- Assign the DLCI 201 to the subinterface on R2

Ask Yourself

- Is a point-to-point subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If not, what other configuration steps are required?
- When DLCIs are learned via LMI from the Frame Relay cloud, where are they assigned?
- How can I change where they are assigned?
- Is the multipoint subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the methods to configure this resolution?

Final Configuration

```
R1:
interface Serial0/0
  encapsulation frame-relay
!
interface Serial0/0.1 point-to-point
ip address 10.0.0.1 255.0.0.0
```

y

```

frame-relay interface-dlci 102

R2:
interface Serial0/0
 encapsulation frame-relay
!
interface Serial0/0.1 multipoint
 ip address 10.0.0.2 255.0.0.0
 frame-relay interface-dlci 201

```

Verification

R1#ping 10.0.0.2

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

```

R1#ping 255.255.255.255 repeat 1

```

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.2, 48 ms

```

R2#ping 255.255.255.255 repeat 1

```

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 8 ms

```

R1#show frame-relay pvc 102

```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

  input pkts 12          output pkts 24          in bytes 1248
  out bytes 5340        dropped pkts 0          in pkts dropped 0
  out pkts dropped 0    out bytes dropped 0
  in FECN pkts 0        inBECN pkts 0          out FECN pkts 0
  out BECN pkts 0        in DE pkts 0           out DE pkts 0
  out bcast pkts 13     out bcast bytes 4196
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:09:11, last time pvc status changed 00:08:33

```

R1#show frame-relay map

```

Serial0/0.1 (up): point-to-point dlci, dlci 102(0x66,0x1860), broadcast
                  status defined, active

```

R2#show frame-relay pvc 201

```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

  input pkts 9          output pkts 8          in bytes 1103

```

y


```
out bytes 762                dropped pkts 0                in pkts dropped 0
out pkts dropped 0           out bytes dropped 0
in FECN pkts 0              in BECN pkts 0              out FECN pkts 0
out BECN pkts 0             in DE pkts 0                out DE pkts 0
out bcst pkts 2             out bcst bytes 138
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:01:07, last time pvc status changed 00:01:07

R2#show frame-relay map
Serial0/0.1 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), dynamic,
                broadcast,, status defined, active
```

Breakdown

In the above example, a point-to-point interface is used on R1, while a multipoint subinterface is used on R2. On R1, layer 3 to layer 2 protocol resolution is not required. Instead, the circuit is simply assigned to the point-to-point subinterface with the `frame-relay interface-dlci` command. On R2, the circuit 102 is assigned to the multipoint subinterface with the `frame-relay interface-dlci` command. Since IP is configured on the multipoint subinterface, a Frame Relay Inverse-ARP request is sent out the circuit for IP, and R1 replies with its IP address 10.0.0.1.

When the configuration is verified with the `show frame-relay map` command, a point-to-point resolution is seen on R1 with broadcast support, while a dynamic unicast layer 3 to layer 2 mapping is seen on R2, also with broadcast support. This configuration dictates that not only is layer 3 to layer 2 resolution only locally significant, the type of interface, whether a main interface, multipoint subinterface, or point-to-point subinterface, is also only locally significant.

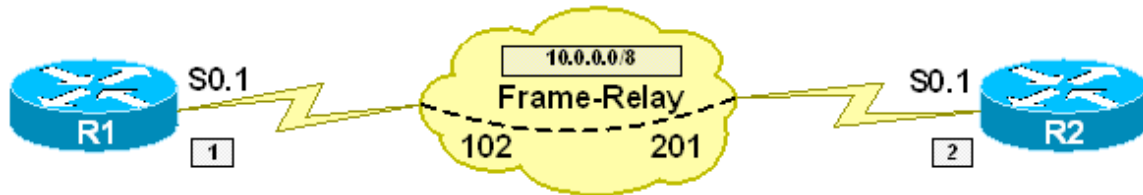


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Point-to-Point Subinterfaces and Multipoint Subinterfaces: Static Mapping

Objective: Configure a Frame Relay PVC connection between R1 and R2 using a point-to-point subinterface on R1 and a multipoint subinterface on R2 using static layer 3 to layer 2 resolution



Directions

- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Create a point-to-point subinterface S0.1 on R1
- Create a multipoint subinterface S0.1 on R2
- Configure the subinterface on R1 with the IP address 10.0.0.1/8
- Configure the subinterface on R2 with the IP address 10.0.0.2/8
- Assign the DLCI 102 to the subinterface on R1
- Configure a static layer 3 to layer 2 resolution on R2's subinterface to use DLCI 201 to reach the IP address 10.0.0.1
- Ensure that broadcast and multicast support is enabled on the circuit

Ask Yourself

- Is a point-to-point subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If not, what other configuration steps are required?
- When DLCIs are learned via LMI from the Frame Relay cloud, where are they assigned?
- How can I change where they are assigned?
- Is the multipoint subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the methods to configure this resolution?

Final Configuration

```

R1:
interface Serial0/0
  encapsulation frame-relay
  !
interface Serial0/0.1 point-to-point
  ip address 10.0.0.1 255.0.0.0
  frame-relay interface-dlci 102

R2:
interface Serial0/0
  encapsulation frame-relay
  !
interface Serial0/0.1 multipoint
  ip address 10.0.0.2 255.0.0.0
  frame-relay map ip 10.0.0.1 201 broadcast

```

Verification

R1#ping 10.0.0.2

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

```

R1#ping 255.255.255.255 repeat 1

```

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.2, 48 ms

```

R2#ping 255.255.255.255 repeat 1

```

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 8 ms

```

R1#show frame-relay pvc 102

```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

  input pkts 12          output pkts 24          in bytes 1248
  out bytes 5340        dropped pkts 0          in pkts dropped 0
  out pkts dropped 0    out bytes dropped 0
  in FECN pkts 0        in BECN pkts 0         out FECN pkts 0
  out BECN pkts 0       in DE pkts 0           out DE pkts 0
  out bcast pkts 13     out bcast bytes 4196
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:09:11, last time pvc status changed 00:08:33

```

R1#show frame-relay map

```

Serial0/0.1 (up): point-to-point dlci, dlci 102(0x66,0x1860), broadcast

```

y

```
status defined, active

R2#show frame-relay pvc 201

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

input pkts 7          output pkts 6          in bytes 965
out bytes 624        dropped pkts 0          in pkts dropped 0
out pkts dropped 0    out bytes dropped 0
in FECN pkts 0       in BECN pkts 0         out FECN pkts 0
out BECN pkts 0       in DE pkts 0           out DE pkts 0
out bcast pkts 1      out bcast bytes 104
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:01:10, last time pvc status changed 00:01:10

R2#show frame-relay map
Serial0/0.1 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), static,
                broadcast,
                CISCO, status defined, active
```

Breakdown

In the above example, a point-to-point interface is used on R1, while a multipoint subinterface used on R2. On R1, layer 3 to layer 2 protocol resolution is not required. Instead, the circuit is simply assigned to the point-to-point subinterface with the `frame-relay interface-dlci` command. On R2, a static layer 3 to layer 2 resolution is configured with the `frame-relay map ip 10.0.0.1 201 broadcast` statement.

When the configuration is verified with the `show frame-relay map` command, a point-to-point resolution is seen on R1 with broadcast support, while a static unicast layer 3 to layer 2 mapping is seen on R2, also with broadcast support. This configuration dictates that not only is layer 3 to layer 2 resolution only locally significant, the type of interface, whether a main interface, multipoint subinterface, or point-to-point subinterface, is also only locally significant.

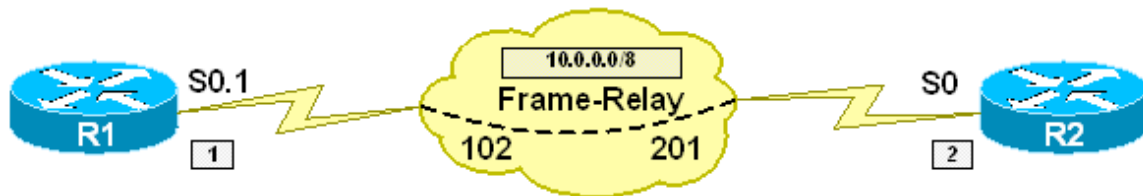


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Main Interface: Inverse-ARP, Multipoint Interface: Inverse-ARP

Objective: Configure a Frame Relay PVC connection between R1 and R2 using a multipoint subinterface on R1 and the main interface on R2, both using Frame Relay Inverse-ARP

**Directions**

- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Create a multipoint subinterface S0.1 on R1
- Configure the subinterface on R1 with the IP address 10.0.0.1/8
- Configure the main interface on R2 with the IP address 10.0.0.2/8
- Assign the DLCI 102 to the subinterface on R1

Ask Yourself

- Is a multipoint subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the different methods to configure this resolution?
- What steps are required to configure dynamic layer 3 to layer 2 resolution via Frame Relay Inverse-ARP?
- When DLCIs are learned via LMI from the Frame Relay cloud, where are they assigned?
- How can I change where they are assigned?

Final Configuration

```

R1:
interface Serial0/0
 encapsulation frame-relay
 !
interface Serial0/0.1 multipoint
 ip address 10.0.0.1 255.0.0.0
 frame-relay interface-dlci 102

R2:
interface Serial0/0
 ip address 10.0.0.2 255.0.0.0
 encapsulation frame-relay

```

Verification

R1#ping 10.0.0.2

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms

```

R1#show frame-relay pvc 102

```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

  input pkts 2           output pkts 1           in bytes 68
  out bytes 34           dropped pkts 0          in pkts dropped 0
  out pkts dropped 0     out bytes dropped 0
  in FECN pkts 0        in BECN pkts 0         out FECN pkts 0
  out BECN pkts 0       in DE pkts 0           out DE pkts 0
  out bcast pkts 1      out bcast bytes 34
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:07:18, last time pvc status changed 00:06:01

```

R1#show frame-relay map

```

Serial0/0.1 (up): ip 10.0.0.2 dlci 102(0x66,0x1860), dynamic,
                  broadcast,, status defined, active

```

R2#show frame-relay pvc 201

```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 6           output pkts 6           in bytes 554
  out bytes 554         dropped pkts 0          in pkts dropped 0
  out pkts dropped 0     out bytes dropped 0
  in FECN pkts 0        in BECN pkts 0         out FECN pkts 0
  out BECN pkts 0       in DE pkts 0           out DE pkts 0
  out bcast pkts 1      out bcast bytes 34
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:08:18, last time pvc status changed 00:07:12

```

y

```
R2#show frame-relay map
Serial0/0 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), dynamic,
                broadcast,, status defined, active
```

Breakdown

In the above example, R1 uses a multipoint subinterface while R2 uses the main interface. Since by default when DLCIs are learned via LMI they are assigned to the main interface, the DLCI on R1 must be moved to the subinterface with the `frame-relay interface-dlci` command. On R2 this command is not required because circuit 201 will be assigned to the main interface by default. Since IP is enabled on both of these interfaces, Frame Relay Inverse-ARP request will be generated by both devices.

To verify that the DLCIs have been assigned to the subinterface on R1 and the main interface on R2, issue the `show frame-relay pvc` command. Like in the previous cases, layer 3 to layer 2 protocol resolution is verified by issuing the `show frame-relay map` command in privilege level mode. This configuration dictates that not only is layer 3 to layer 2 resolution only locally significant, the type of interface, whether a main interface, multipoint subinterface, or point-to-point subinterface, is also only locally significant.

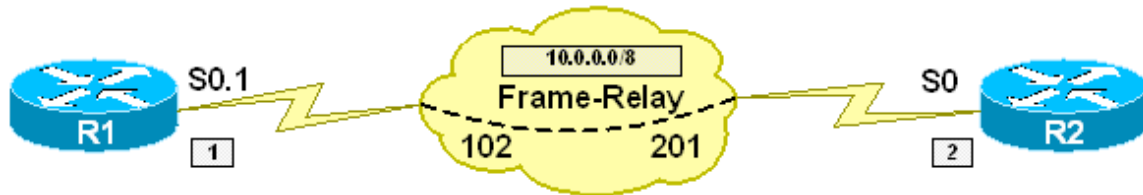


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Main Interface: Inverse-ARP, Multipoint Interface: Static Mapping

Objective: Configure a Frame Relay PVC connection between R1 and R2 using a multipoint subinterface on R1 with Frame Relay Inverse-ARP and the main interface on R2 using static layer 3 to layer 2 resolution



Directions

- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Create a multipoint subinterface S0.1 on R1
- Configure the subinterface on R1 with the IP address 10.0.0.1/8
- Configure the main interface on R2 with the IP address 10.0.0.2/8
- Assign the DLCI 102 to the subinterface on R1
- Configure a static layer 3 to layer 2 resolution on R2 to reach the IP address 10.0.0.1 via DLCI 201
- Ensure that broadcast and multicast traffic can be sent across the circuit

Ask Yourself

- Is a multipoint subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the different methods to configure this resolution?
- What steps are required to configure dynamic layer 3 to layer 2 resolution via Frame Relay Inverse-ARP?
- When DLCIs are learned via LMI from the Frame Relay cloud, where are they assigned?
- How can I change where they are assigned to?
- What steps are required to configure static layer 3 to layer 2 resolution?

Final Configuration

```
R1:
interface Serial0/0
  encapsulation frame-relay
  !
interface Serial0/0.1 multipoint
  ip address 10.0.0.1 255.0.0.0
```

y


```
frame-relay interface-dlci 102
```

R2:

```
interface Serial0/0
ip address 10.0.0.2 255.0.0.0
encapsulation frame-relay
frame-relay map ip 10.0.0.1 201 broadcast
```

Verification

R1#ping 10.0.0.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms

R1#show frame-relay pvc 102

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

```
input pkts 2          output pkts 1          in bytes 68
out bytes 34          dropped pkts 0         in pkts dropped 0
out pkts dropped 0    out bytes dropped 0
in FECN pkts 0        in BECN pkts 0        out FECN pkts 0
out BECN pkts 0        in DE pkts 0          out DE pkts 0
out bcast pkts 1      out bcast bytes 34
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:07:18, last time pvc status changed 00:06:01
```

R1#show frame-relay map

```
Serial0/0.1 (up): ip 10.0.0.2 dlci 102(0x66,0x1860), dynamic,
                broadcast,, status defined, active
```

R2#show frame-relay pvc 201

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

```
input pkts 6          output pkts 6          in bytes 554
out bytes 554          dropped pkts 0         in pkts dropped 0
out pkts dropped 0    out bytes dropped 0
in FECN pkts 0        in BECN pkts 0        out FECN pkts 0
out BECN pkts 0        in DE pkts 0          out DE pkts 0
out bcast pkts 1      out bcast bytes 34
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:08:18, last time pvc status changed 00:07:12
```

R2#show frame-relay map

```
Serial0/0 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), static,
                broadcast,
                CISCO, status defined, active
```

Breakdown

In the above example, R1 uses a multipoint subinterface while R2 uses the main interface. Since by default when DLCIs are learned via LMI they are assigned to the main interface, the DLCI on R1 must be moved to the subinterface with the `frame-relay interface-dlci` command. On R2 this command is not required because circuit 201 will be assigned to the main interface by default. Since IP is enabled on R1's subinterface, a Frame Relay Inverse-ARP request will be generated by out circuit 102 to discover R2's IP address 10.0.0.2. On R2's main interface, layer 3 to layer 2 resolution is configured statically with the `frame-relay map ip 10.0.0.1 201 broadcast` command.

To verify that the DLCIs have been assigned to the subinterface on R1 and the main interface on R2, issue the `show frame-relay pvc` command. Like in the previous cases, layer 3 to layer 2 protocol resolution is verified by issuing the `show frame-relay map` command in privilege level mode. This configuration dictates that not only is layer 3 to layer 2 resolution only locally significant, the type of interface, whether a main interface, multipoint subinterface, or point-to-point subinterface, is also only locally significant.

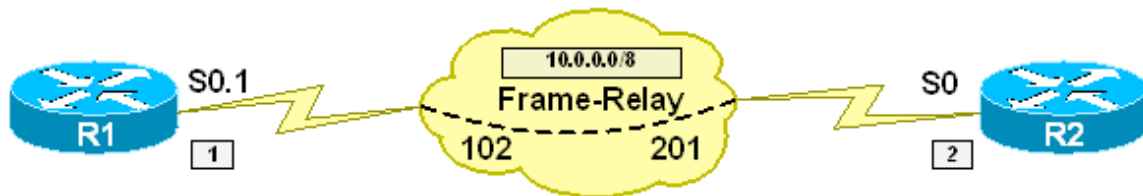


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Main Interface: Static Mapping, Multipoint Interface: Inverse-ARP

Objective: Configure a Frame Relay PVC connection between R1 and R2 using a multipoint subinterface on R1 using Frame Relay Inverse-ARP and the main interface on R2 using static layer 3 to layer 2 resolution



Directions

- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Create a multipoint subinterface S0.1 on R1
- Configure the subinterface on R1 with the IP address 10.0.0.1/8
- Configure the main interface on R2 with the IP address 10.0.0.2/8
- Assign the DLCI 102 to the subinterface on R1
- Configure static layer 3 to layer 2 resolution on R2 to use DLCI 201 to reach R1's IP address of 10.0.0.1
- Ensure that broadcast and multicast traffic can be sent over the circuit

Ask Yourself

- Is a multipoint subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the different methods to configure this resolution?
- What steps are required to configure dynamic layer 3 to layer 2 resolution via Frame Relay Inverse-ARP?
- When DLCIs are learned via LMI from the Frame Relay cloud, where are they assigned?
- How can I change where they are assigned?
- What steps are required to configure static layer 3 to layer 2 resolution?

Final Configuration

```
R1:
interface Serial0/0
  encapsulation frame-relay
!
interface Serial0/0.1 multipoint
  ip address 10.0.0.1 255.0.0.0
  frame-relay interface-dlci 102
```

y

```

R2:
interface Serial0/0
 ip address 10.0.0.2 255.0.0.0
 encapsulation frame-relay
 frame-relay map ip 10.0.0.1 201 broadcast

```

Verification

R1#ping 10.0.0.2

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms

```

R1#show frame-relay pvc 102

```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

  input pkts 2          output pkts 1          in bytes 68
  out bytes 34         dropped pkts 0         in pkts dropped 0
  out pkts dropped 0   out bytes dropped 0
  in FECN pkts 0       in BECN pkts 0        out FECN pkts 0
  out BECN pkts 0     in DE pkts 0          out DE pkts 0
  out bcast pkts 1    out bcast bytes 34
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:07:18, last time pvc status changed 00:06:01

```

R1#show frame-relay map

```

Serial0/0.1 (up): ip 10.0.0.2 dlci 102(0x66,0x1860), dynamic,
                  broadcast,, status defined, active

```

R2#show frame-relay pvc 201

```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 6          output pkts 6          in bytes 554
  out bytes 554        dropped pkts 0         in pkts dropped 0
  out pkts dropped 0   out bytes dropped 0
  in FECN pkts 0       in BECN pkts 0        out FECN pkts 0
  out BECN pkts 0     in DE pkts 0          out DE pkts 0
  out bcast pkts 1    out bcast bytes 34
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:08:18, last time pvc status changed 00:07:12

```

R2#show frame-relay map

```

Serial0/0 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), static,
                  broadcast,
                  CISCO, status defined, active

```

Breakdown

In the above example, R1 uses a multipoint subinterface while R2 uses the main interface. Since by default when DLCIs are learned via LMI they are assigned to the main interface, the DLCI on R1 must be moved to the subinterface with the `frame-relay interface-dlci` command. On R2 this command is not required because circuit 201 will be assigned to the main interface by default. On R1, since IP is enabled on the interface, a Frame Relay Inverse-ARP request will be generated to resolve R2's IP address. On R2's main interface, layer 3 to layer 2 resolution is configured statically with the `frame-relay map ip 10.0.0.1 201 broadcast` command.

To verify that the DLCIs have been assigned to the subinterface on R1 and the main interface on R2, issue the `show frame-relay pvc` command. Like in the previous cases, layer 3 to layer 2 protocol resolution is verified by issuing the `show frame-relay map` command in privilege level mode. This configuration dictates that not only is layer 3 to layer 2 resolution only locally significant, the type of interface, whether a main interface, multipoint subinterface, or point-to-point subinterface, is also only locally significant.

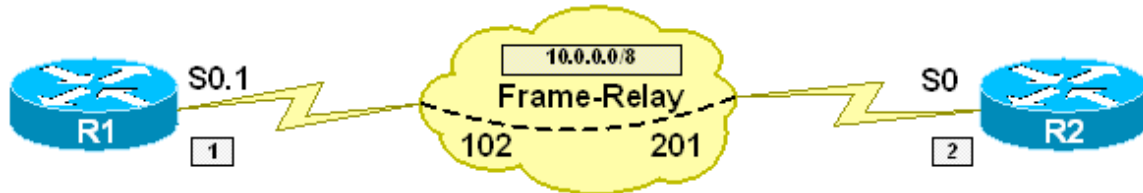


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Main Interface: Static Mapping, Multipoint Interface: Static Mapping

Objective: Configure a Frame Relay PVC connection between R1 and R2 using a multipoint subinterface on R1 and the main interface on R2, both using static layer 3 to layer 2 resolution



Directions

- Configure Frame Relay encapsulation on R1 and R2's main serial interfaces
- Create a multipoint subinterface S0.1 on R1
- Configure the subinterface on R1 with the IP address 10.0.0.1/8
- Configure the main interface on R2 with the IP address 10.0.0.2/8
- Configure a static layer 3 to layer 2 resolution on R1 to reach the IP address 10.0.0.2 via DLCI 102
- Configure a static layer 3 to layer 2 resolution on R2 to reach the IP address 10.0.0.1 via DLCI 201
- Ensure that broadcast and multicast traffic can be sent across the circuit

Ask Yourself

- Is a multipoint subinterface running Frame Relay multipoint or point-to-point?
- Does this interface require layer 3 to layer 2 protocol resolution?
- If so, what are the different methods to configure this resolution?
- What steps are required to configure static layer 3 to layer 2 resolution?
- When DLCIs are learned via LMI from the Frame Relay cloud, where are they assigned?
- How can I change where they are assigned?

Final Configuration

```
R1:
interface Serial0/0
  encapsulation frame-relay
!
interface Serial0/0.1 multipoint
  ip address 10.0.0.1 255.0.0.0
  frame-relay map ip 10.0.0.2 102 broadcast

R2:
interface Serial0/0
```

y

```
ip address 10.0.0.2 255.0.0.0
encapsulation frame-relay
frame-relay map ip 10.0.0.1 201 broadcast
```

Verification

R1#ping 10.0.0.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms

R1#show frame-relay pvc 102

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

```
input pkts 2          output pkts 1          in bytes 68
out bytes 34         dropped pkts 0         in pkts dropped 0
out pkts dropped 0   out bytes dropped 0
in FECN pkts 0      in BECN pkts 0        out FECN pkts 0
out BECN pkts 0     in DE pkts 0          out DE pkts 0
out bcast pkts 1    out bcast bytes 34
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:07:18, last time pvc status changed 00:06:01
```

R1#show frame-relay map

```
Serial0/0.1 (up): ip 10.0.0.2 dlci 102(0x66,0x1860), static,
broadcast,
CISCO, status defined, active
```

R2#show frame-relay pvc 201

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

```
input pkts 6          output pkts 6          in bytes 554
out bytes 554        dropped pkts 0         in pkts dropped 0
out pkts dropped 0   out bytes dropped 0
in FECN pkts 0      in BECN pkts 0        out FECN pkts 0
out BECN pkts 0     in DE pkts 0          out DE pkts 0
out bcast pkts 1    out bcast bytes 34
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:08:18, last time pvc status changed 00:07:12
```

R2#show frame-relay map

```
Serial0/0 (up): ip 10.0.0.1 dlci 201(0xC9,0x3090), static,
broadcast,
CISCO, status defined, active
```

Breakdown

In the above example, R1 uses a multipoint subinterface while R2 uses the main interface. By default when DLCIs are learned via LMI they are assigned to the main interface. The DLCI 102 on R1 is moved to the subinterface with the `frame-relay map ip 10.0.0.2 102 broadcast` command. This statement also manually resolves R2's IP address to R1's local layer 2 address. On R2's main interface, layer 3 to layer 2 resolution is configured statically with the `frame-relay map ip 10.0.0.1 201 broadcast` command.

To verify that the DLCIs have been assigned to the subinterface on R1 and the main interface on R2, issue the `show frame-relay pvc` command. Like in the previous cases, layer 3 to layer 2 protocol resolution is verified by issuing the `show frame-relay map` command in privilege level mode. This configuration dictates that not only is layer 3 to layer 2 resolution only locally significant, the type of interface, whether a main interface, multipoint subinterface, or point-to-point subinterface, is also only locally significant.

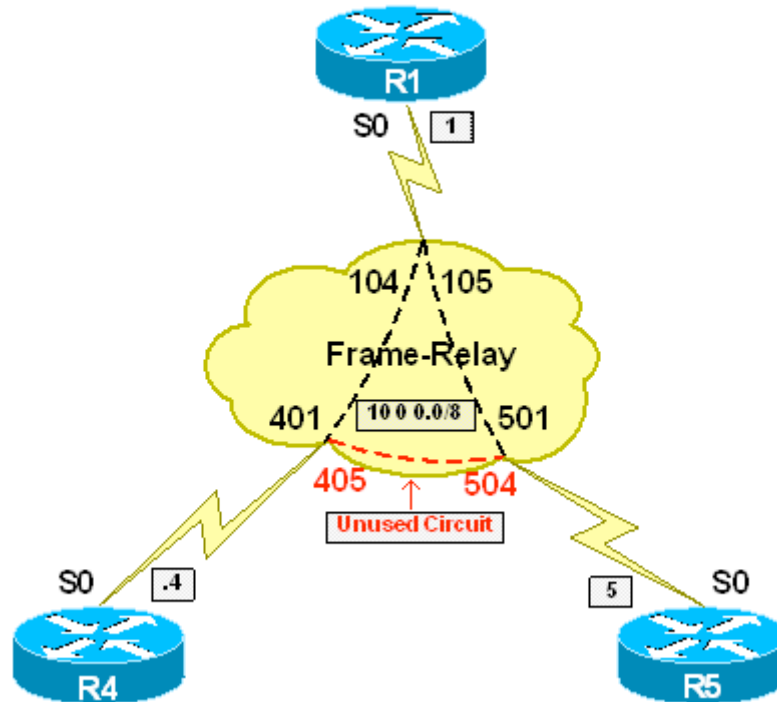


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Hub-and-Spoke, Main Interfaces with Inverse-ARP

Objective: Configure a Frame Relay Hub-and-Spoke network between R1, R4, and R5 with R1 as the hub, using main interfaces and Frame Relay Inverse-ARP on all endpoints



Directions

- Configure R1's main serial interface with the IP address 10.0.0.1/8
- Configure R4's main serial interface with the IP address 10.0.0.4/8
- Configure R5's main serial interface with the IP address 10.0.0.5/8
- Configure Frame Relay encapsulation on R1, R4, and R5's main serial interfaces
- There is an unused circuit provisioned between R4 and R5; Do not allow protocol resolution to occur on this circuit via Frame Relay Inverse-ARP
- Verify reachability between all endpoints of the network

Ask Yourself

- When R1 receives an Inverse-ARP request from R4 and R5, will it forward it on?
- What are the implications of this?
- How can I selectively disable Inverse-ARP on a specific DLCI?

Final Configuration**R1:**

```
interface Serial0/0
 ip address 10.0.0.1 255.0.0.0
 encapsulation frame-relay
```

R4:

```
interface Serial0/0
 ip address 10.0.0.4 255.0.0.0
 encapsulation frame-relay
 no frame-relay inverse-arp ip 405
```

R5:

```
interface Serial0/0
 ip address 10.0.0.5 255.0.0.0
 encapsulation frame-relay
 no frame-relay inverse-arp ip 504
```

Verification**R1#show frame-relay map**

```
Serial0/0 (up): ip 10.0.0.4 dlci 104(0x68,0x1880), dynamic,
                broadcast,, status defined, active
Serial0/0 (up): ip 10.0.0.5 dlci 105(0x69,0x1890), dynamic,
                broadcast,, status defined, active
```

R4#show frame-relay map

```
Serial0/0 (up): ip 10.0.0.1 dlci 401(0x191,0x6410), dynamic,
                broadcast,, status defined, active
```

R5#show frame-relay map

```
Serial0/0 (up): ip 10.0.0.1 dlci 501(0x1F5,0x7C50), dynamic,
                broadcast,, status defined, active
```

R1#ping 10.0.0.4

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms
```

R1#ping 10.0.0.5

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms
```

R4#ping 10.0.0.5

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.5, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

R4#debug frame-relay packet

```
Frame Relay packet debugging is on
```

```
R4#ping 10.0.0.5 repeat 1
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 10.0.0.5, timeout is 2 seconds:
.
Success rate is 0 percent (0/1)
Serial0/0:Encaps failed--no map entry link 7(IP)
```

Breakdown

Partial mesh NBMA refers to an NBMA network that does not have layer 2 circuits provisioned between all endpoints of the network. One common configuration of partial mesh is known as hub-and-spoke. In a hub-and-spoke topology one central device, or hub, has layer 2 circuits provisioned to all other devices on the network, or spokes, but the spokes do not have layer 2 circuits provisioned between each other. In order for traffic to pass between the spokes, it must first transit the hub.

In the above example, R1 is the hub for R4 and R5, with circuits provisioned between R1 & R4, and R1 & R5. In addition to this, a circuit has been provisioned between R4 and R5 in the Frame Relay cloud, however it will not be used in the configuration. In a production network, it is highly unlikely that additional circuits will be learned from the Frame Relay cloud via LMI that are not actually used. However, in the case of the CCIE lab exam, the Frame Relay switch used may have additional circuits provisioned in order to ease in designing flexible network topologies for multiple tests. Therefore, while an unlikely issue in a real network, dealing with unused circuits will play a significant role in the many partial meshed configurations that will be covered in this lab series.

In the above configuration, Frame Relay encapsulation is enabled, along with IP addresses, on the main serial interfaces connecting to the Frame Relay cloud on R1, R4, and R5. As seen in previous examples, this will dictate that Frame Relay Inverse-ARP requests will be sent out all circuits learned via LMI. Output from the `show frame-relay pvc | include DLCI` command shows that additional circuits have been learned from the cloud than are used in the topology. Therefore, Frame Relay Inverse-ARP has been selectively disabled on the circuit between R4 and R5 with the `no frame-relay inverse-arp ip 405` and `no frame-relay inverse-arp ip 504` commands. Note that this configuration only affects the specified DLCI and protocol pairs.

Once Frame Relay Inverse-ARP requests have been sent, it is seen from the `show frame-relay map` output that R1 has resolution for R4 and R5, but R4 and R5 only have resolution for R1. This is due to the fact that a Frame Relay Inverse-ARP request can only resolve the layer 3 address of a device that is directly connected at layer 2. In other words, R4 and R5 will not have reachability to each other, as protocol resolution has not occurred for each others addresses.

y

This can be seen from the `debug frame-relay packet` output on R4 that says `Encaps failed--no map entry link 7(IP)`. This means that R4 is trying to encapsulate a unicast IP packet on the interface for which it does not have layer 3 to layer 2 resolution for. Methods of resolving this issue will be explored in the further labs of this series.

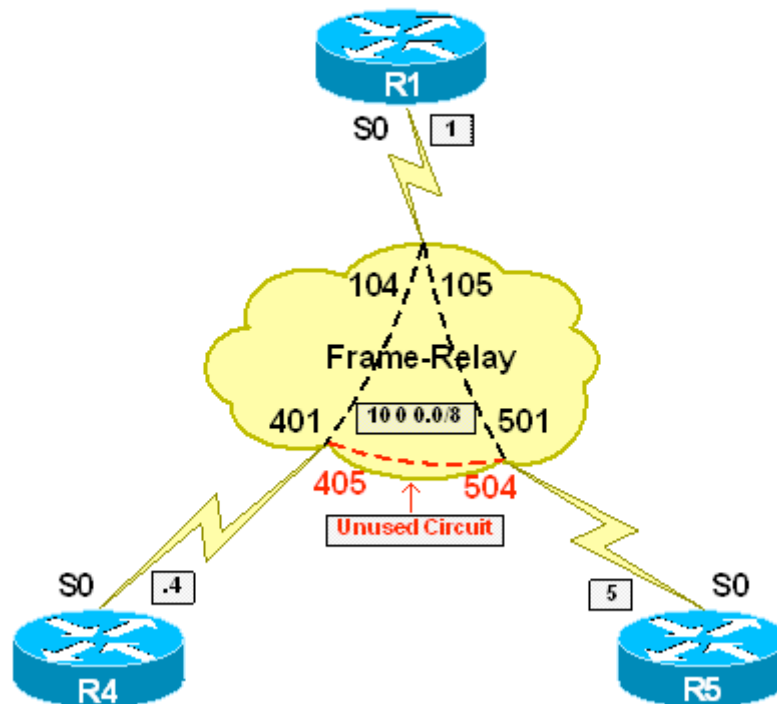


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Hub-and-Spoke, Main Interfaces w/ Inverse-ARP & Static Mappings

Objective: Configure a Frame Relay Hub-and-Spoke network between R1, R4, and R5 with R1 as the hub, using main interfaces on all devices, Frame Relay Inverse-ARP on R1, and static layer 3 to layer 2 resolution on R4 and R5



Directions

- Configure R1's main serial interface with the IP address 10.0.0.1/8
- Configure R4's main serial interface with the IP address 10.0.0.4/8
- Configure R5's main serial interface with the IP address 10.0.0.5/8
- Configure Frame Relay encapsulation on R1, R4, and R5's main serial interfaces
- Disable Frame Relay Inverse-ARP on R4 and R5
- Configure static layer 3 to layer 2 resolution on R4 and R5 for all endpoints of the network
- Ensure that broadcast and multicast traffic can be sent over the circuits, but do not send redundant broadcast traffic from the spokes to the hub
- Verify reachability between all endpoints of the network

Ask Yourself

y

- If R4 and R5 disable Frame Relay Inverse-ARP, will this affect R1?
- What circuit does R4 use to connect to R1? to R5?
- How will this affect layer 3 to layer 2 resolution?
- What circuit does R5 use to connect to R1? to R4?
- How will this affect layer 3 to layer 2 resolution?

Final Configuration

R1:

```
interface Serial0/0
 ip address 10.0.0.1 255.0.0.0
 encapsulation frame-relay
```

R4:

```
interface Serial0/0
 ip address 10.0.0.4 255.0.0.0
 encapsulation frame-relay
 frame-relay map ip 10.0.0.1 401 broadcast
 frame-relay map ip 10.0.0.5 401
 no frame-relay inverse-arp
```

R5:

```
interface Serial0/0
 ip address 10.0.0.5 255.0.0.0
 encapsulation frame-relay
 frame-relay map ip 10.0.0.1 501 broadcast
 frame-relay map ip 10.0.0.4 501
 no frame-relay inverse-arp
```

Verification

R1#show frame-relay map

```
Serial0/0 (up): ip 10.0.0.4 dlci 104(0x68,0x1880), dynamic,
                broadcast,, status defined, active
Serial0/0 (up): ip 10.0.0.5 dlci 105(0x69,0x1890), dynamic,
                broadcast,, status defined, active
```

R4#show frame-relay map

```
Serial0/0 (up): ip 10.0.0.1 dlci 401(0x191,0x6410), static,
                broadcast,
                CISCO, status defined, active
Serial0/0 (up): ip 10.0.0.5 dlci 401(0x191,0x6410), static,
                CISCO, status defined, active
```

R5#show frame-relay map

```
Serial0/0 (up): ip 10.0.0.1 dlci 501(0x1F5,0x7C50), static,
                broadcast,
                CISCO, status defined, active
Serial0/0 (up): ip 10.0.0.4 dlci 501(0x1F5,0x7C50), static,
                CISCO, status defined, active
```

R1#ping 10.0.0.4

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms
```

y

```
R1#ping 10.0.0.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms

R1#ping 255.255.255.255 repeat 1
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.5, 16 ms
Reply to request 0 from 10.0.0.4, 16 ms

R4#ping 10.0.0.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms

R4#ping 255.255.255.255 repeat 1
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 4 ms

R5#ping 10.0.0.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/12 ms

R5#ping 255.255.255.255 repeat 1
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 72 ms
```

Breakdown

Partial mesh NBMA refers to an NBMA network that does not have layer 2 circuits provisioned between all endpoints of the network. One common configuration of partial mesh is known as hub-and-spoke. In a hub-and-spoke topology one central device, or hub, has layer 2 circuits provisioned to all other devices on the network, or spokes, but the spokes do not have layer 2 circuits provisioned between each other. In order for traffic to pass between the spokes, it must first transit the hub.

In the above example, R1 connects to R4 and R5 via circuits 104 and 105 respectively. R4 and R5 connect to R1 via circuits 401 and 501 respectively. All

y

interfaces in the configuration are main interfaces, and therefore require layer 3 to layer 2 resolution. On R1, this is accomplished automatically through Frame Relay Inverse-ARP. On R4 and R5, Frame Relay Inverse-ARP is disabled for all circuits and protocols with the `no frame-relay inverse-arp` command, and static `frame-relay map` statements are configured. Disabling Inverse-ARP on these devices ensures that they will not create dynamic resolution for each other out the unused circuit that is provisioned between them.

Note that on R4 and R5, multiple mapping statements resolve to the same layer 2 address. This is due to the fact that in order for traffic to pass between R4 and R5, it must first transit R1. Also, note that the `broadcast` keyword on the end of their `frame-relay map` statements is only used on one mapping. This prevents the spokes from duplicating the same broadcast or multicast packet that is sent to the interface out the same circuit. Note that the mapping that the `broadcast` keyword is associated with is arbitrary, and could be moved to the mapping for R4 and R5's addresses respectively. This is due to the fact that the broadcast resolution does not relate to the unicast address, but instead only relates to the layer 2 circuit address.

This configuration is verified both through by viewing resolution through the `show frame-relay map` command, and by sending both unicast and broadcast ICMP traffic across the circuits.

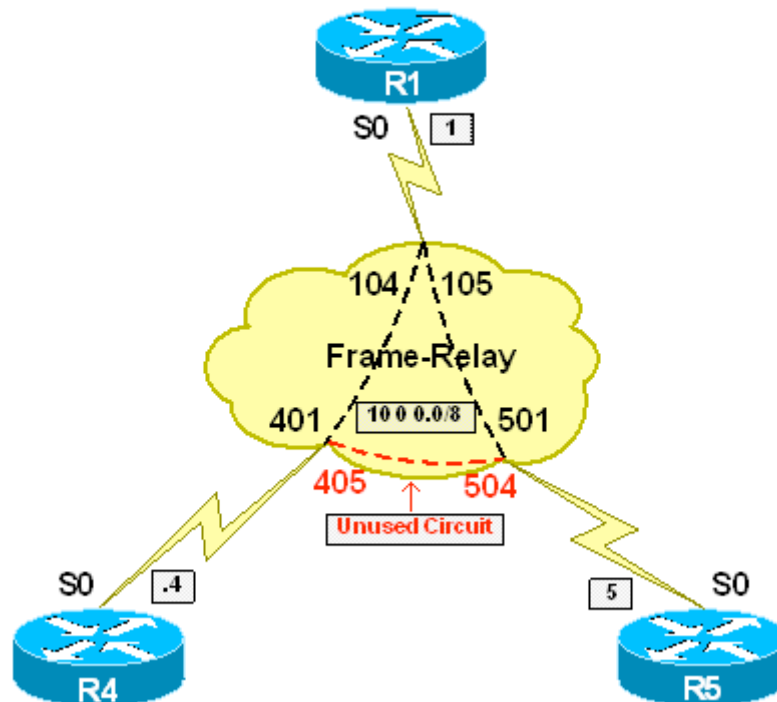


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Hub-and-Spoke, Main Interfaces w/ Static Mappings

Objective: Configure a Frame Relay Hub-and-Spoke network between R1, R4, and R5 with R1 as the hub, using main interfaces and static layer 3 to layer 2 resolution on all devices



Directions

- Configure R1's main serial interface with the IP address 10.0.0.1/8
- Configure R4's main serial interface with the IP address 10.0.0.4/8
- Configure R5's main serial interface with the IP address 10.0.0.5/8
- Configure Frame Relay encapsulation on R1, R4, and R5's main serial interfaces
- Disable Frame Relay Inverse-ARP on R4 and R5
- Configure static layer 3 to layer 2 resolution on R1, R4, and R5 for all endpoints of the network
- Ensure that broadcast and multicast traffic can be sent over the circuits, but do not send redundant broadcast traffic from the spokes to the hub
- Verify reachability between all endpoints of the network

Ask Yourself

- What circuit does R4 use to connect to R1? to R5?
- How will this affect layer 3 to layer 2 resolution?

- What circuit does R5 use to connect to R1? to R4?
- How will this affect layer 3 to layer 2 resolution?

Final Configuration

```

R1:
interface Serial0/0
 ip address 10.0.0.1 255.0.0.0
 encapsulation frame-relay
 frame-relay map ip 10.0.0.4 104 broadcast
 frame-relay map ip 10.0.0.5 105 broadcast

R4:
interface Serial0/0
 ip address 10.0.0.4 255.0.0.0
 encapsulation frame-relay
 frame-relay map ip 10.0.0.1 401 broadcast
 frame-relay map ip 10.0.0.5 401
 no frame-relay inverse-arp

R5:
interface Serial0/0
 ip address 10.0.0.5 255.0.0.0
 encapsulation frame-relay
 frame-relay map ip 10.0.0.1 501 broadcast
 frame-relay map ip 10.0.0.4 501
 no frame-relay inverse-arp

```

Verification

```

R1#show frame-relay map
Serial0/0 (up): ip 10.0.0.4 dlci 104(0x68,0x1880), static,
                broadcast,
                CISCO, status defined, active
Serial0/0 (up): ip 10.0.0.5 dlci 105(0x69,0x1890), static,
                broadcast,
                CISCO, status defined, active

R4#show frame-relay map
Serial0/0 (up): ip 10.0.0.1 dlci 401(0x191,0x6410), static,
                broadcast,
                CISCO, status defined, active
Serial0/0 (up): ip 10.0.0.5 dlci 401(0x191,0x6410), static,
                CISCO, status defined, active

R5#show frame-relay map
Serial0/0 (up): ip 10.0.0.1 dlci 501(0x1F5,0x7C50), static,
                broadcast,
                CISCO, status defined, active
Serial0/0 (up): ip 10.0.0.4 dlci 501(0x1F5,0x7C50), static,
                CISCO, status defined, active

R1#ping 10.0.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

```

y

```
R1#ping 10.0.0.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms

R1#ping 255.255.255.255 repeat 1
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.5, 16 ms
Reply to request 0 from 10.0.0.4, 16 ms

R4#ping 10.0.0.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms

R4#ping 255.255.255.255 repeat 1
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 4 ms

R5#ping 10.0.0.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/12 ms

R5#ping 255.255.255.255 repeat 1
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 72 ms
```

Breakdown

Partial mesh NBMA refers to an NBMA network that does not have layer 2 circuits provisioned between all endpoints of the network. One common configuration of partial mesh is known as hub-and-spoke. In a hub-and-spoke topology one central device, or hub, has layer 2 circuits provisioned to all other devices on the network, or spokes, but the spokes do not have layer 2 circuits provisioned between each other. In order for traffic to pass between the spokes, it must first transit the hub.

In the above example, R1 connects to R4 and R5 via circuits 104 and 105 respectively. R4 and R5 connect to R1 via circuits 401 and 501 respectively. All

interfaces in the configuration are main interfaces, and therefore require layer 3 to layer 2 resolution. On all routers this is accomplished manually through the `frame-relay map` statements. Additionally, on R4 and R5, Frame Relay Inverse-ARP is disabled for all circuits and protocols with the `no frame-relay inverse-arp` command. Disabling Inverse-ARP on these devices ensures that they will not create dynamic resolution for each other out the unused circuit that is provisioned between them.

Note that on R4 and R5, multiple mapping statements resolve to the same layer 2 address. This is due to the fact that in order for traffic to pass between R4 and R5, it must first transit R1. Also, note that the `broadcast` keyword on the end of their `frame-relay map` statements is only used on one mapping. This prevents the spokes from duplicating the same broadcast or multicast packet that is sent to the interface out the same circuit. Note that the mapping that the `broadcast` keyword is associated with is arbitrary, and could be moved to the mapping for R4 and R5's addresses respectively. This is due to the fact that the broadcast resolution does not relate to the unicast address, but instead only relates to the layer 2 circuit address.

This configuration is verified both through by viewing resolution through the `show frame-relay map` command, and by sending both unicast and broadcast ICMP traffic across the circuits.

Recommended Reading

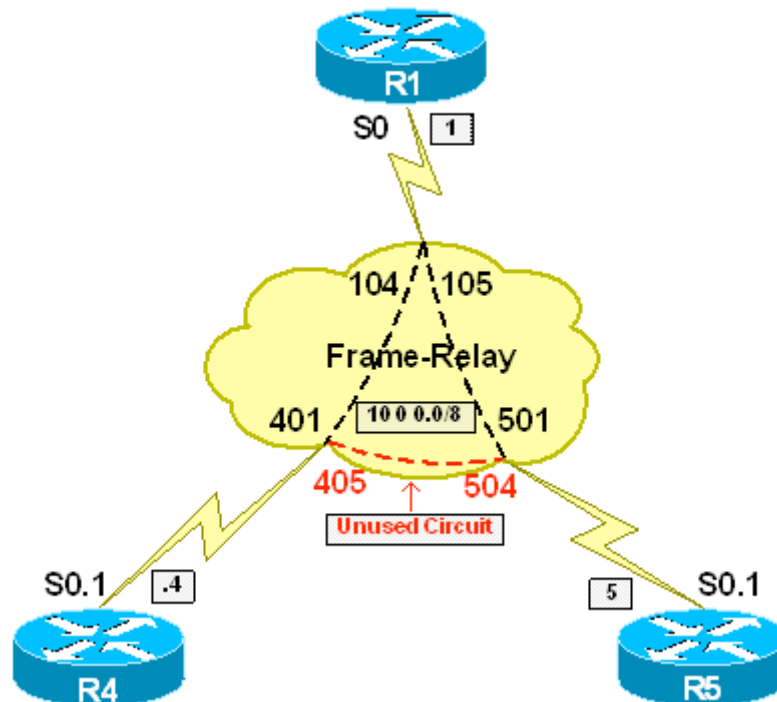


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Hub-and-Spoke, Main Interfaces w/ Inverse-ARP & Point-to-Point Subinterfaces

Objective: Configure a Frame Relay Hub-and-Spoke network between R1, R4, and R5 with R1 as the hub, using main interfaces on R1 with Frame Relay Inverse-ARP and point-to-point subinterfaces on R4 and R5



Directions

- Configure Frame Relay encapsulation on R1, R4, and R5's main serial interfaces
- Create point-to-point subinterfaces S0.1 on R4 and R5
- Configure R1's main serial interface with the IP address 10.0.0.1/8
- Configure R4's subinterface with the IP address 10.0.0.4/8
- Configure R5's subinterface with the IP address 10.0.0.5/8
- Assign the DLCI 401 to R4's subinterface
- Assign the DLCI 501 to R5's subinterface
- Verify reachability between all endpoints of the network

Ask Yourself

- Is the main interface on R1 multipoint or point-to-point?
- Does this interface require protocol resolution?
- Do the point-to-point interfaces on R4 and R5 require protocol resolution?
- What implications will this configuration have on reachability?

Final Configuration

```

R1:
interface Serial0/0
 ip address 10.0.0.1 255.0.0.0
 encapsulation frame-relay

R4:
interface Serial0/0
 encapsulation frame-relay
!
interface Serial0/0.1 point-to-point
 ip address 10.0.0.4 255.0.0.0
 frame-relay interface-dlci 401

R5:
interface Serial0/0
 encapsulation frame-relay
!
interface Serial0/0.1 point-to-point
 ip address 10.0.0.5 255.0.0.0
 frame-relay interface-dlci 501

```

Verification

```

R1#show frame-relay map
Serial0/0 (up): ip 10.0.0.4 dlci 104(0x68,0x1880), dynamic,
                broadcast,, status defined, active
Serial0/0 (up): ip 10.0.0.5 dlci 105(0x69,0x1890), dynamic,
                broadcast,, status defined, active

R4#show frame-relay map
Serial0/0.1 (up): point-to-point dlci, dlci 401(0x191,0x6410), broadcast
                  status defined, active

R5#show frame-relay map
Serial0/0.1 (up): point-to-point dlci, dlci 501(0x1F5,0x7C50), broadcast
                  status defined, active

R1#ping 10.0.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

R1#ping 10.0.0.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

R1#ping 255.255.255.255 repeat 1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

```

```
Reply to request 0 from 10.0.0.4, 77 ms
Reply to request 0 from 10.0.0.5, 77 ms

R4#ping 10.0.0.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/10/12 ms

R4#ping 255.255.255.255 repeat 1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 65 ms

R5#ping 255.255.255.255 repeat 1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 8 ms
```

Breakdown

Partial mesh NBMA refers to an NBMA network that does not have layer 2 circuits provisioned between all endpoints of the network. One common configuration of partial mesh is known as hub-and-spoke. In a hub-and-spoke topology one central device, or hub, has layer 2 circuits provisioned to all other devices on the network, or spokes, but the spokes do not have layer 2 circuits provisioned between each other. In order for traffic to pass between the spokes, it must first transit the hub.

In the above example, R1 connects to R4 and R5 via circuits 104 and 105 respectively. R4 and R5 connect to R1 via circuits 401 and 501 respectively. R1's interface is the main interfaces, and therefore require layer 3 to layer 2 resolution. This is accomplished dynamically through Frame Relay Inverse-ARP, which is enabled for IP as soon as Frame Relay is encapsulated and an IP address is assigned. R4 and R5 connect to the Frame Relay cloud using point-to-point subinterfaces, which do not required layer 3 to layer 2 resolution. As long as the circuit is assigned to the interface with the `frame-relay interface-dlci` command, all traffic destined for the interface will use the assigned DLCI in the packet. Note that Frame Relay Inverse-ARP does not need to be disabled on R4 and R5, as the unused circuit between them is assigned to the main interface when it is learned via LMI, and there is no IP address on the main interface. This implies that Frame Relay Inverse-ARP for IP will not be enabled on the circuit.

This exercise shows that both interface types and layer 3 to layer 2 resolution is only locally significant. This configuration is verified both through by viewing

resolution through the `show frame-relay map` command, and by sending both unicast and broadcast ICMP traffic across the circuits.

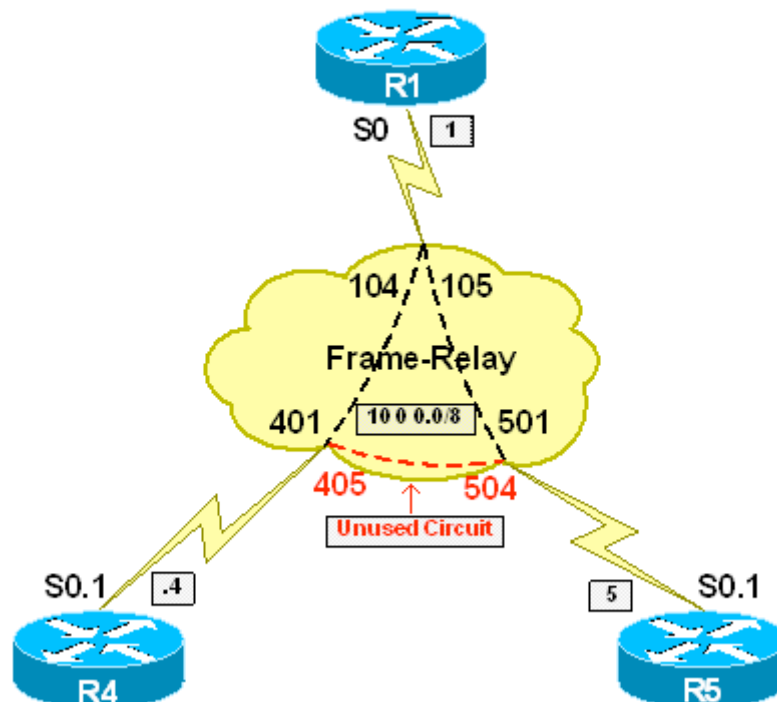


Recommended Reading

[Configuring Frame Relay](#)

Frame-Relay Hub-and-Spoke, Main Interfaces w/ Static Mappings & Point-to-Point Subinterfaces

Objective: Configure a Frame Relay Hub-and-Spoke network between R1, R4, and R5 with R1 as the hub, using main interfaces on R1 with static layer 3 to layer 2 resolutions and point-to-point subinterfaces on R4 and R5



Directions

- Configure Frame Relay encapsulation on R1, R4, and R5's main serial interfaces
- Create point-to-point subinterfaces S0.1 on R4 and R5
- Configure R1's main serial interface with the IP address 10.0.0.1/8
- Configure R4's subinterface with the IP address 10.0.0.4/8
- Configure R5's subinterface with the IP address 10.0.0.5/8
- Configure static layer 3 to layer 2 resolution on R1 for both R4 and R5
- Assign the DLCI 401 to R4's subinterface
- Assign the DLCI 501 to R5's subinterface
- Verify reachability between all endpoints of the network, including the support for broadcast and multicast transmission

Ask Yourself

y

- Is the main interface on R1 multipoint or point-to-point?
- Does this interface require protocol resolution?
- Do the point-to-point interfaces on R4 and R5 require protocol resolution?
- What implications will this configuration have on reachability?

Final Configuration

```
R1:
interface Serial0/0
 ip address 10.0.0.1 255.0.0.0
 encapsulation frame-relay
 frame-relay map ip 10.0.0.4 104 broadcast
 frame-relay map ip 10.0.0.5 105 broadcast

R4:
interface Serial0/0
 encapsulation frame-relay
 !
interface Serial0/0.1 point-to-point
 ip address 10.0.0.4 255.0.0.0
 frame-relay interface-dlci 401

R5:
interface Serial0/0
 encapsulation frame-relay
 !
interface Serial0/0.1 point-to-point
 ip address 10.0.0.5 255.0.0.0
 frame-relay interface-dlci 501
```

Verification

```
R1#show frame-relay map
Serial0/0 (up): ip 10.0.0.4 dlci 104(0x68,0x1880), static,
                broadcast,
                CISCO, status defined, active
Serial0/0 (up): ip 10.0.0.5 dlci 105(0x69,0x1890), static,
                broadcast,
                CISCO, status defined, active

R4#show frame-relay map
Serial0/0.1 (up): point-to-point dlci, dlci 401(0x191,0x6410), broadcast
                  status defined, active

R5#show frame-relay map
Serial0/0.1 (up): point-to-point dlci, dlci 501(0x1F5,0x7C50), broadcast
                  status defined, active

R1#ping 10.0.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

R1#ping 10.0.0.5
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

R1#ping 255.255.255.255 repeat 1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.4, 77 ms
Reply to request 0 from 10.0.0.5, 77 ms

R4#ping 10.0.0.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/10/12 ms

R4#ping 255.255.255.255 repeat 1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 65 ms

R5#ping 255.255.255.255 repeat 1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

Reply to request 0 from 10.0.0.1, 8 ms
```

Breakdown

Partial mesh NBMA refers to an NBMA network that does not have layer 2 circuits provisioned between all endpoints of the network. One common configuration of partial mesh is known as hub-and-spoke. In a hub-and-spoke topology one central device, or hub, has layer 2 circuits provisioned to all other devices on the network, or spokes, but the spokes do not have layer 2 circuits provisioned between each other. In order for traffic to pass between the spokes, it must first transit the hub.

In the above example, R1 connects to R4 and R5 via circuits 104 and 105 respectively. R4 and R5 connect to R1 via circuits 401 and 501 respectively. R1's interface is the main interfaces, and therefore require layer 3 to layer 2 resolution. This is accomplished manually through `frame-relay map` statements. R4 and R5 connect to the Frame Relay cloud using point-to-point subinterfaces, which do not required layer 3 to layer 2 resolution. As long as the circuit is assigned to the interface with the `frame-relay interface-dlci` command, all traffic destined for the interface will use the assigned DLCI in the packet. Note that Frame Relay Inverse-ARP does not need to be disabled on R4 and R5, as the unused circuit between them is assigned to the main interface when it is learned

y

via LMI, and there is no IP address on the main interface. This implies that Frame Relay Inverse-ARP for IP will not be enabled on the circuit.

This exercise shows that both interface types and layer 3 to layer 2 resolution is only locally significant. This configuration is verified both through by viewing resolution through the `show frame-relay map` command, and by sending both unicast and broadcast ICMP traffic across the circuits.



Recommended Reading

[Configuring Frame Relay](#)

