

Copyright Information

Copyright © 2008 Internetwork Expert, Inc. All rights reserved.

The following publication, CCIE R&S Lab Workbook Volume I Version 5.0, 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 R&S Lab Workbook Volume I Version 5.0, 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.

Table of Contents

Frame Relay	1
2.1 Inverse-ARP	1
2.2 Static Mappings	1
2.3 Point-to-Point Subinterfaces	1
2.4 Multipoint Subinterfaces & Inverse-ARP	2
2.5 Multipoint Subinterfaces & Static Mappings	2
2.6 Hub-and-Spoke & Static Mappings	2
2.7 Hub-and-Spoke & Inverse-ARP	3
2.8 Hub-and-Spoke & Point-to-Point Subinterfaces	3
2.9 Disabling Inverse-ARP	4
2.10 Back-to-Back Frame Relay	4
2.11 Frame Relay End-to-End Keepalives	5
2.12 Frame Relay Broadcast Queue	5
2.13 Frame Relay TCP & RTP Header Compression	5
2.13 PPP over Frame Relay	5
2.14 Bridging over Frame Relay	6
Frame Relay Solutions	7
2.1 Inverse-ARP	7
2.2 Static Mappings	10
2.3 Point-to-Point Subinterfaces	12
2.4 Multipoint Subinterfaces & Inverse-ARP	14
2.5 Multipoint Subinterfaces & Static Mappings	17
2.6 Hub-and-Spoke & Static Mappings	20
2.7 Hub-and-Spoke & Inverse-ARP	24
2.8 Hub-and-Spoke & Point-to-Point Subinterfaces	27
2.9 Disabling Inverse-ARP	30
2.10 Back-to-Back Frame Relay	34
2.11 Frame Relay End-to-End Keepalives	36
2.12 Frame Relay Broadcast Queue	38
2.13 Frame Relay TCP & RTP Header Compression	39
2.13 PPP over Frame Relay	40
2.14 Bridging over Frame Relay	43

Frame Relay

 **Note**

Erase and reload all devices to a blank configuration prior to starting.

2.1 Inverse-ARP

- Configure the IP address of R6's Serial interface using the information in the diagram.
- Enable Frame Relay on this link.
- Ensure R6 learns an IP to DLCI mapping dynamically for PVC 101 connected to BB1, and that these devices have unicast, multicast, and broadcast reachability to each other.

2.2 Static Mappings

- Configure the IP addresses of R1 and R5's Serial interfaces connected to the Frame Relay cloud using the information in the diagram.
- Enable Frame Relay on this link and configure static IP to DLCI mappings so that R1 and R5 have unicast, multicast, and broadcast reachability to each other.

2.3 Point-to-Point Subinterfaces

- Enable Frame Relay on R2's link connecting to the Frame Relay cloud.
- Create a point-to-point subinterface numbered 205 on R2 with the IP address 155.X.25.2/24 and assign DLCI 205 to it.
- Create a point-to-point subinterface numbered 502 on R5 with the IP address 155.X.25.5/24 and assign DLCI 502 to it.
- Ensure that these devices have unicast, multicast, and broadcast reachability to each other on this segment.

2.4 Multipoint Subinterfaces & Inverse-ARP

- Enable Frame Relay on R3 and R4's connections to the Frame Relay cloud.
- Create a multipoint subinterface numbered 100 on R3 with the IP address 155.X.100.3/24 and assign DLCIs 304 and 305 to it.
- Create a multipoint subinterface numbered 100 on R4 with the IP address 155.X.100.4/24 and assign DLCIs 403 and 405 to it.
- Create a multipoint subinterface numbered 100 on R5 with the IP address 155.X.100.5/24 and assign DLCIs 503 and 504 to it.
- Ensure that R3, R4, and R5 all learn dynamic IP to DLCI mappings for each other on this segment and have unicast, multicast, and broadcast reachability to each other.

2.5 Multipoint Subinterfaces & Static Mappings

- Modify the previous configuration between R3, R4, and R5 to use static IP to DLCI mappings as opposed to dynamic mappings.
- Ensure that these devices have unicast, multicast, and broadcast reachability to each other on this segment.

 **Note**

Erase and reload all devices to a blank configuration before continuing.

2.6 Hub-and-Spoke & Static Mappings

- Enable Frame Relay on the Serial interfaces of R1, R3, and R5 connected to the Frame Relay switch.
- Assign IP addresses to these interfaces per the diagram.
- Configure static IP to DLCI mappings on these devices to gain reachability to each other using only the DLCI assignments shown in the diagram.
- Ensure that R1 & R5 and R3 & R5 have unicast, multicast, and broadcast reachability to each other, and that unicast traffic between R1 & R3 transits through R5.

2.7 Hub-and-Spoke & Inverse-ARP

- Reset the Frame Relay interfaces of R1, R3, and R5 to its default configuration.
- Enable Frame Relay on the Serial interfaces of R1, R3, and R5 connected to the Frame Relay switch.
- Assign IP addresses to these interfaces per the diagram.
- Configure the network in such a way that R1 & R5 and R3 & R5 resolve each other's IP addresses dynamically through Inverse-ARP, while R1 and R3 resolve each other's addresses statically.
- Ensure that R1 & R5 and R3 & R5 have unicast, multicast, and broadcast reachability to each other, that unicast traffic between R1 & R3 transits through R5.

2.8 Hub-and-Spoke & Point-to-Point Subinterfaces

- Reset the Frame Relay interfaces of R1, R3, and R5 to its default configuration.
- Enable Frame Relay on the Serial interfaces of R1, R3, and R5 connected to the Frame Relay switch.
- Configure a point-to-point subinterface numbered 105 on R1 and assign it the IP address 155.X.0.1/24 & DLCI 105.
- Configure a point-to-point subinterface numbered 305 on R3 and assign it the IP address 155.X.0.3/24 & DLCI 305.
- Assign the IP address 155.X.0.5 to R5's main Serial interface and configure static IP to DLCI mappings for R1 and R3.
- Ensure that R1 & R5 and R3 & R5 have unicast, multicast, and broadcast reachability to each other, that unicast traffic between R1 & R3 transits through R5.

 **Note**

Erase and reload all devices to a blank configuration before continuing.

2.9 Disabling Inverse-ARP

- Enable Frame Relay on R1, R2, R3, R4, and R5's interfaces connected to the Frame Relay cloud.
- Configure R1's interface with the IP address 155.X.100.1/24.
- Configure R2's interface with the IP address 155.X.100.2/24.
- Configure R3's interface with the IP address 155.X.200.3/24.
- Configure R4's interface with the IP address 155.X.200.4/24.
- Create a multipoint subinterface numbered 100 on R5 with the IP address 155.X.100.5/24 and the DLCIs 501 & 502.
- Create a multipoint subinterface numbered 200 on R5 with the IP address 155.X.200.5/24 and the DLCIs 503 & 504.
- Configure R1 and R2 so that they will not send Inverse-ARP requests for IP out any DLCIs learned from the Frame Relay cloud other than those in the diagram.
- Configure R3 and R4 to disable Inverse-ARP on DLCIs learned from the Frame Relay cloud that are not listed in the diagram by assigning them to the subinterface 999.
- Ensure that R5 has unicast, multicast, and broadcast reachability to R1, R2, R3, and R4.
- R1 and R2 should be able to reach R5, but not each other.
- R3 and R4 should be able to reach R5, but not each other.
- R1, R2, R3, and R4 should not be able to reach each other.

2.10 Back-to-Back Frame Relay

- Configure IP addresses on the directly connected Serial link between R4 and R5 using the information in the diagram.
- Configure Frame Relay PVC 514 on this link without using LMI.
- Configure static IP to DLCI mappings on R4 and R5 to obtain reachability on this segment.

2.11 Frame Relay End-to-End Keepalives

- Create a Frame Relay map-class called FREEK on R4 and R5.
- Configure R4 to respond to Frame Relay keepalives, and R5 to request them.
- Set R5 to poll every 5 seconds with an event window of 10 and an error threshold of 8.
- R4 should apply this configuration to its main Serial0/0 interface, while R5 should apply this just to the DLCI connecting to R4.

2.12 Frame Relay Broadcast Queue

- Tune the Frame Relay broadcast queue size between R4 and R5 to 100 packets.

2.13 Frame Relay TCP & RTP Header Compression

- Enable RTP header compression for DLCIs 503 and 305 on R5 and R3 respectively.
- Enable TCP header compression for DLCIs 501 and 105 on R5 and R1 respectively.

 **Note**

Erase and reload all devices to a blank configuration before continuing.

2.13 PPP over Frame Relay

- Create Virtual-Template interfaces on R1 and R5 using the IP addresses 155.X.0.1/24 and 155.X.0.5/24 respectively.
- Enable Frame Relay on R1 and R5's Serial interfaces attached to the Frame Relay cloud, and attach the Virtual-Templates to the DLCIs connecting these devices.
- Ensure that R1 and R5 have unicast, multicast, and broadcast reachability to each other over this segment.

2.14 Bridging over Frame Relay

- Remove the previous PPPoFR configuration on R5.
- Configure ports Fa0/2 and Fa0/24 on SW2 in VLAN 100.
- Configure SW1's interface Fa0/5 with the IP address 192.10.X.7/24.
- Enable Frame Relay on the Serial interfaces connecting to the Frame Relay cloud on R2 and R5.
- Disable IP routing on R2 and R5 and remove any IP addresses from their interfaces.
- Configure an IEEE STP bridge group numbered 1 on R2 & R5, and apply this to their Frame Relay connections and their FastEthernet0/0 interfaces.
- Ensure that SW1 and BB2 have unicast, multicast, and broadcast reachability to each other.

Frame Relay Solutions

2.1 Inverse-ARP

- Configure the IP address of R6's Serial interface using the information in the diagram.
- Enable Frame Relay on this link.
- Ensure R6 learns an IP to DLCI mapping dynamically for PVC 101 connected to BB1, and that these devices have unicast, multicast, and broadcast reachability to each other.

Configuration

```
R6:  
interface Serial0/0  
  encapsulation frame-relay  
  ip address 54.1.1.6 255.255.255.0
```

Verification

Note

Frame-Relay is a non-broadcast multi-access (NBMA) media, which means that there is no layer 2 address that maps to all devices in the network, like FFFF:FFFF:FFFF does in Ethernet. This means that for interfaces that can have multiple layer 2 destinations, i.e. the multipoint main interface or multipoint subinterface, layer 3 to layer 2 resolution must be performed either manually with the **frame-relay map command** or automatically through Frame Relay Inverse-ARP.

Once the **encapsulation frame-relay** command is issued at the interface level the router will learn all DLCIs provisioned by the service provider through LMI. Without other configuration all DLCIs will be automatically assigned to the main interface when learned. The first verification for this is to issue the **show frame-relay pvc** command. If the PVC status is *active*, then frames can be sent and received over it.

```
Rack1R6#show frame-relay pvc 101
```

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

```
DLCI = 101, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0
```

```

input pkts 218          output pkts 25          in bytes 24901
out bytes 1802         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 3     out bcast bytes 102
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:10:21, last time pvc status changed 00:01:10
```

Once a layer 3 protocol is configured on the interface, such as IPv4, Inverse-ARP requests will automatically be sent out all DLCIs learned via LMI. If the device on the other end of the link is running that protocol stack, a reply will be received indicating their layer 3 address that is assigned on that circuit. Verification for this can be performed with the **show frame-relay map** command. Note that the *broadcast* keyword appears in the below show output. This indicates that Inverse-ARP automatically supports the sending of broadcast and multicast frames as replicated unicasts out the mapped circuit.

```
Rack1R6#show frame-relay map
```

```

Serial0/0 (up): ip 54.1.1.254 dlci 101(0x65,0x1850), dynamic,
                broadcast,, status defined, active
Serial0/0 (up): ip 54.1.2.254 dlci 100(0x64,0x1840), dynamic,
                broadcast,, status defined, active
Serial0/0 (up): ip 54.1.3.254 dlci 51(0x33,0xC30), dynamic,
                broadcast,, status defined, active
```

To verify that transport is achieved it's a good idea at this point to test both unicast and multicast/broadcast reachability. Unicast can be tested with a simple ICMP ping, while multicast/broadcast transport can be tested by pinging the all hosts broadcast address of 255.255.255.255.

```
Rack1R6#ping 54.1.1.254
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 54.1.1.254, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
```

```
Rack1R6#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 54.1.2.254, 76 ms
```

```
Reply to request 0 from 54.1.1.254, 108 ms
```

```
Reply to request 0 from 54.1.3.254, 92 ms
```

Pitfall

Note that R6 learns dynamic mappings for IP addresses not on the same subnet as configured on the interface. This is because multiple PVCs are learned from the Frame Relay network through LMI, and Inverse-ARP requests are sent out all of them. The result of this is seen when broadcast traffic is sent out the interface. Even though R6 does not have the subnet 54.1.3.0/24 assigned it received a reply from the broadcast request sent out to that circuit. To fix this behavior Inverse-ARP should be disabled on the other circuits (not PVC 101), or PVC 101 should be assigned to a subinterface. The ideal design solution would be to use a subinterface to avoid this problem.

2.2 Static Mappings

- Configure the IP addresses of R1 and R5's Serial interfaces connected to the Frame Relay cloud using the information in the diagram.
- Enable Frame Relay on this link and configure static IP to DLCI mappings so that R1 and R5 have unicast, multicast, and broadcast reachability to each other.

Configuration

```
R1:  
interface Serial0/0  
  encapsulation frame-relay  
  ip address 155.1.0.1 255.255.255.0  
  frame-relay map ip 155.1.0.5 105 broadcast
```

```
R5:  
interface Serial0/0  
  encapsulation frame-relay  
  ip address 155.1.0.5 255.255.255.0  
  frame-relay map ip 155.1.0.1 501 broadcast
```

Verification

Note

Static Frame Relay mappings override any dynamically learned Inverse-ARP mappings. If multipoint interfaces are used in your configuration, i.e. the main interface or a multipoint subinterface, static **frame-relay map** commands would be the ideal design solution, as sometimes the Inverse-ARP protocol is unreliable and dynamic mappings disappear. Note that broadcast and multicast transport is not automatically enabled unless the *broadcast* keyword is added at the end of the **frame-relay map** command.

```
Rack1R1#show frame-relay map
Serial0/0 (up): ip 155.1.0.5 dlci 105(0x69,0x1890), static,
                broadcast,
                CISCO, status defined, active
```

```
Rack1R5#show frame-relay map
Serial0/0 (up): ip 155.1.0.1 dlci 501(0x1F5,0x7C50), static,
                broadcast,
                CISCO, status defined, active
```

Just like using dynamic mappings it's a good idea at this point to test both unicast and multicast/broadcast reachability. Unicast can be tested with a simple ICMP ping, while multicast/broadcast transport can be tested by pinging the all hosts broadcast address of 255.255.255.255.

```
Rack1R1#ping 155.1.0.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.0.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms
```

```
Rack1R1#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 155.1.0.5, 116 ms
```

```
Rack1R5#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 155.1.0.1, 60 ms
```

2.3 Point-to-Point Subinterfaces

- Enable Frame Relay on R2's link connecting to the Frame Relay cloud.
- Create a point-to-point subinterface numbered 205 on R2 with the IP address 155.X.25.2/24 and assign DLCI 205 to it.
- Create a point-to-point subinterface numbered 502 on R5 with the IP address 155.X.25.5/24 and assign DLCI 502 to it.
- Ensure that these devices have unicast, multicast, and broadcast reachability to each other on this segment.

Configuration

```
R2:
interface Serial0/0
  encapsulation frame-relay
!
interface Serial0/0.205 point-to-point
  ip address 155.1.25.2 255.255.255.0
  frame-relay interface-dlci 205

R5:
interface Serial0/0
  encapsulation frame-relay
!
interface Serial0/0.502 point-to-point
  ip address 155.1.25.5 255.255.255.0
  frame-relay interface-dlci 502
```


Verification

Note

Point-to-point Frame Relay interfaces, such as point-to-point subinterfaces or PPP over Frame Relay interfaces, do not require layer 3 to layer 2 resolution because there is only one possible layer 2 destination reachable out the link. As long as the circuit is assigned with the **frame-relay interface-dlci** command all unicast, multicast, or broadcast traffic for the configured layer 3 protocol can be encapsulated.

This type of configuration would be the ideal design solution for running Frame Relay, as complex dynamic or static Frame Relay mappings are not required. Using point-to-point interfaces will also solve many other layer 3 design issues with protocols such as OSPF, Multicast, and IPv6. These issues are covered in depth in those particular protocol chapters of the workbook.

Rack1R2#show frame-relay map

```
Serial0/0.205 (up): point-to-point dlci, dlci 205(0xCD,0x30D0),  
broadcast  
                status defined, active
```

Rack1R5#show frame-relay map

```
Serial0/0 (up): ip 155.1.0.1 dlci 501(0x1F5,0x7C50), static,  
                broadcast,  
                CISCO, status defined, active  
Serial0/0.502 (up): point-to-point dlci, dlci 502(0x1F6,0x7C60),  
broadcast  
                status defined, active
```

Rack1R2#ping 155.1.25.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.25.5, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

Rack1R2#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 155.1.25.5, 128 ms

2.4 Multipoint Subinterfaces & Inverse-ARP

- Enable Frame Relay on R3 and R4's connections to the Frame Relay cloud.
- Create a multipoint subinterface numbered 100 on R3 with the IP address 155.X.100.3/24 and assign DLCIs 304 and 305 to it.
- Create a multipoint subinterface numbered 100 on R4 with the IP address 155.X.100.4/24 and assign DLCIs 403 and 405 to it.
- Create a multipoint subinterface numbered 100 on R5 with the IP address 155.X.100.5/24 and assign DLCIs 503 and 504 to it.
- Ensure that R3, R4, and R5 all learn dynamic IP to DLCI mappings for each other on this segment and have unicast, multicast, and broadcast reachability to each other.

Configuration

```
R3:
interface Serial1/0
 encapsulation frame-relay
 !
interface Serial1/0.100 multipoint
 ip address 155.1.100.3 255.255.255.0
 frame-relay interface-dlci 304
 frame-relay interface-dlci 305

R4:
interface Serial0/0
 encapsulation frame-relay
 !
interface Serial0/0.100 multipoint
 ip address 155.1.100.4 255.255.255.0
 frame-relay interface-dlci 403
 frame-relay interface-dlci 405

R5:
interface Serial0/0
 encapsulation frame-relay
 !
interface Serial0/0.100 multipoint
 ip address 155.1.100.5 255.255.255.0
 frame-relay interface-dlci 503
 frame-relay interface-dlci 504
```

Verification

Note

Like the main interface, multipoint subinterfaces require layer 3 to layer 2 resolution since there are multiple possible layer 2 destinations. Typically this configuration is done with the static **frame-relay map** commands, however it can also be done dynamically with Inverse-ARP like in this example. By assigning the DLCI to the interface with the **frame-relay interface-dlci** command, Inverse-ARP request will automatically be sent out the PVCs for the configured protocol stack, IPv4.

Rack1R3#show frame-relay map

```
Serial1/0.100 (up): ip 155.1.100.4 dlci 304(0x130,0x4C00), dynamic,
                    broadcast,, status defined, active
Serial1/0.100 (up): ip 155.1.100.5 dlci 305(0x131,0x4C10), dynamic,
                    broadcast,, status defined, active
```

Rack1R4#show frame-relay map

```
Serial0/0.100 (up): ip 155.1.100.3 dlci 403(0x193,0x6430), dynamic,
                    broadcast,, status defined, active
Serial0/0.100 (up): ip 155.1.100.5 dlci 405(0x195,0x6450), dynamic,
                    broadcast,, status defined, active
```

Rack1R5#show frame-relay map

```
Serial0/0.100 (up): ip 155.1.100.3 dlci 503(0x1F7,0x7C70), dynamic,
                    broadcast,, status defined, active
Serial0/0.100 (up): ip 155.1.100.4 dlci 504(0x1F8,0x7C80), dynamic,
                    broadcast,, status defined, active
Serial0/0 (up): ip 155.1.0.1 dlci 501(0x1F5,0x7C50), static,
                    broadcast,
                    CISCO, status defined, active
Serial0/0.502 (up): point-to-point dlci, dlci 502(0x1F6,0x7C60),
broadcast
                    status defined, active
```

Once the mappings appear testing unicast and broadcast reachability can eliminate and higher layer problems, such as incorrect IP address assignments.

```
Rack1R3#ping 155.1.100.4
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 155.1.100.4, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/59/60 ms
```

```
Rack1R3#ping 155.1.100.5
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 155.1.100.5, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/59/60 ms
```

```
Rack1R3#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 155.1.100.5, 60 ms
```

```
Reply to request 0 from 155.1.100.4, 72 ms
```

2.5 Multipoint Subinterfaces & Static Mappings

- Modify the previous configuration between R3, R4, and R5 to use static IP to DLCI mappings as opposed to dynamic mappings.
- Ensure that these devices have unicast, multicast, and broadcast reachability to each other on this segment.

Configuration

```
R3:
interface Serial1/0
  encapsulation frame-relay
  !
interface Serial1/0.100 multipoint
  ip address 155.1.100.3 255.255.255.0
  frame-relay map ip 155.1.100.4 304 broadcast
  frame-relay map ip 155.1.100.5 305 broadcast

R4:
interface Serial0/0
  encapsulation frame-relay
  !
interface Serial0/0.100 multipoint
  ip address 155.1.100.4 255.255.255.0
  frame-relay map ip 155.1.100.3 403 broadcast
  frame-relay map ip 155.1.100.5 405 broadcast

R5:
interface Serial0/0
  encapsulation frame-relay
  !
interface Serial0/0.100 multipoint
  ip address 155.1.100.5 255.255.255.0
  frame-relay map ip 155.1.100.3 503 broadcast
  frame-relay map ip 155.1.100.4 504 broadcast
```

Verification

Note

Identical in operation to the previous task, using static **frame-relay map** commands on the multipoint subinterface eliminates any possibility of errors in dynamic negotiation. Ensure to add the *broadcast* keyword at the end of the mapping to ensure that broadcast and multicast packets can be encapsulated.

Rack1R3#show frame-relay map

```
Serial1/0.100 (up): ip 155.1.100.4 dlci 304(0x130,0x4C00), static,
    broadcast,
    CISCO, status defined, active
Serial1/0.100 (up): ip 155.1.100.5 dlci 305(0x131,0x4C10), static,
    broadcast,
    CISCO, status defined, active
```

Rack1R4#show frame-relay map

```
Serial0/0.100 (up): ip 155.1.100.3 dlci 403(0x193,0x6430), static,
    broadcast,
    CISCO, status defined, active
Serial0/0.100 (up): ip 155.1.100.5 dlci 405(0x195,0x6450), static,
    broadcast,
    CISCO, status defined, active
```

Rack1R5#show frame-relay map

```
Serial0/0.100 (up): ip 155.1.100.3 dlci 503(0x1F7,0x7C70), static,
    broadcast,
    CISCO, status defined, active
Serial0/0.100 (up): ip 155.1.100.4 dlci 504(0x1F8,0x7C80), static,
    broadcast,
    CISCO, status defined, active
Serial0/0 (up): ip 155.1.0.1 dlci 501(0x1F5,0x7C50), static,
    broadcast,
    CISCO, status defined, active
Serial0/0.502 (up): point-to-point dlci, dlci 502(0x1F6,0x7C60),
broadcast
    status defined, active
```

Rack1R3#ping 155.1.100.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.100.4, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 57/59/60 ms

Rack1R3#ping 155.1.100.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.100.5, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 60/60/60 ms

Rack1R3#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 155.1.100.5, 72 ms

Reply to request 0 from 155.1.100.4, 88 ms

2.6 Hub-and-Spoke & Static Mappings

- Enable Frame Relay on the Serial interfaces of R1, R3, and R5 connected to the Frame Relay switch.
- Assign IP addresses to these interfaces per the diagram.
- Configure static IP to DLCI mappings on these devices to gain reachability to each other using only the DLCI assignments shown in the diagram.
- Ensure that R1 & R5 and R3 & R5 have unicast, multicast, and broadcast reachability to each other, and that unicast traffic between R1 & R3 transits through R5.

Configuration

```
R1:
interface Serial0/0
  encapsulation frame-relay
  ip address 155.1.0.1 255.255.255.0
  frame-relay map ip 155.1.0.5 105 broadcast
  frame-relay map ip 155.1.0.3 105
```

```
R3:
interface Serial1/0
  encapsulation frame-relay
  ip address 155.1.0.3 255.255.255.0
  frame-relay map ip 155.1.0.5 305 broadcast
  frame-relay map ip 155.1.0.1 305
```

```
R5:
interface Serial0/0
  encapsulation frame-relay
  ip address 155.1.0.5 255.255.255.0
  frame-relay map ip 155.1.0.1 501 broadcast
  frame-relay map ip 155.1.0.3 503 broadcast
```


Verification

Note

Hub-and-spoke, or partial mesh networks, implies that the layer 3 network does not map directly to a full mesh of layer 2 circuits. In this particular design R1, R3, and R5 are all in the subnet 155.1.0.0/24, but there is no layer 2 DLCI directly between R1 and R3. This implies that traffic from R1 to R3 must first transit R5. To accomplish this R1 has a static **frame-relay map** command for the IP addresses of R3 and R5, both reachable through the DLCI to R5. Likewise R3's mappings to both R1 and R5 use the circuit address to R5.

Note that the *broadcast* keyword on R1 and R3 is only associated with one of the mappings. This is to prevent excess broadcast or multicast traffic from being replicated multiple times at layer 2 and going out the same circuit. For example if the broadcast keyword was listed on all mappings in this design, every broadcast packet R1 sends to the interface to be delivered out DLCI 105 would be copied twice, and R5 would receive two packets. The excess traffic will be dropped on R5, and should not cause any reachability problems, but it will result in excess utilization on the link between the devices.

Rack1R1#show frame-relay map

```
Serial0/0 (up): ip 155.1.0.3 dlci 105(0x69,0x1890), static,  
                CISCO, status defined, active  
Serial0/0 (up): ip 155.1.0.5 dlci 105(0x69,0x1890), static,  
                broadcast,  
                CISCO, status defined, active
```

Rack1R3#show frame-relay map

```
Serial1/0 (up): ip 155.1.0.1 dlci 305(0x131,0x4C10), static,  
                CISCO, status defined, active  
Serial1/0 (up): ip 155.1.0.5 dlci 305(0x131,0x4C10), static,  
                broadcast,  
                CISCO, status defined, active
```

Rack1R5#show frame-relay map

```
Serial0/0 (up): ip 155.1.0.1 dlci 501(0x1F5,0x7C50), static,  
                broadcast,  
                CISCO, status defined, active  
Serial0/0 (up): ip 155.1.0.3 dlci 503(0x1F7,0x7C70), static,  
                broadcast,  
                CISCO, status defined, active
```

At this point R1, R3, and R5 should all have unicast reachability to each other, but R1 and R3 will not have broadcast or multicast reachability. This is due to the fact that R5 cannot route broadcasts or link local multicasts between interfaces, which is the desired behavior.

Rack1R1#ping 155.1.0.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 112/116/124 ms

Rack1R1#ping 155.1.0.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.5, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/61 ms

Rack1R1#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 155.1.0.5, 61 ms

Rack1R3#ping 155.1.0.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 116/117/124 ms

Rack1R3#ping 155.1.0.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.5, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 57/59/60 ms

Rack1R3#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 155.1.0.5, 104 ms

Rack1R5#ping 155.1.0.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

Rack1R5#ping 155.1.0.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/59/61 ms

Rack1R5#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 155.1.0.3, 92 ms

Reply to request 0 from 155.1.0.1, 108 ms

2.7 Hub-and-Spoke & Inverse-ARP

- Reset the Frame Relay interfaces of R1, R3, and R5 to its default configuration.
- Enable Frame Relay on the Serial interfaces of R1, R3, and R5 connected to the Frame Relay switch.
- Assign IP addresses to these interfaces per the diagram.
- Configure the network in such a way that R1 & R5 and R3 & R5 resolve each other's IP addresses dynamically through Inverse-ARP, while R1 and R3 resolve each other's addresses statically.
- Ensure that R1 & R5 and R3 & R5 have unicast, multicast, and broadcast reachability to each other, that unicast traffic between R1 & R3 transits through R5.

Configuration

```
R1:
interface Serial0/0
  encapsulation frame-relay
  ip address 155.1.0.1 255.255.255.0
  frame-relay map ip 155.1.0.3 105
```

```
R3:
interface Serial1/0
  encapsulation frame-relay
  ip address 155.1.0.3 255.255.255.0
  frame-relay map ip 155.1.0.1 305
```

```
R5:
interface Serial0/0
  encapsulation frame-relay
  ip address 155.1.0.5 255.255.255.0
```

Verification **Note**

Frame Relay Inverse-ARP packets are link-local to the layer 2 process, and cannot be forwarded between different circuits. This means that when R1 sends an Inverse-ARP request out to R5 on DLCI 105, R5 cannot turn around and forward this request to R3 on DLCI 503. The result of this is that R5 can resolve R1 and R3 via Inverse-ARP, R1 and R3 can resolve R5 via Inverse-ARP, but R1 and R3 cannot resolve each other through Inverse-ARP. To gain reachability between these devices a static **frame-relay map** command must be used to populate the layer 3 to layer 2 resolution table.

Rack1R1#show frame-relay map

```
Serial0/0 (up): ip 155.1.0.3 dlci 105(0x69,0x1890), static,
                CISCO, status defined, active
Serial0/0 (up): ip 155.1.0.5 dlci 105(0x69,0x1890), dynamic,
                broadcast,, status defined, active
```

Rack1R3#show frame-relay map

```
Serial1/0 (up): ip 155.1.0.1 dlci 305(0x131,0x4C10), static,
                CISCO, status defined, active
Serial1/0 (up): ip 155.1.0.5 dlci 305(0x131,0x4C10), dynamic,
                broadcast,, status defined, active
```

Rack1R5#show frame-relay map

```
Serial0/0 (up): ip 155.1.0.1 dlci 501(0x1F5,0x7C50), dynamic,
                broadcast,, status defined, active
Serial0/0 (up): ip 155.1.0.3 dlci 503(0x1F7,0x7C70), dynamic,
                broadcast,, status defined, active
```

Rack1R1#ping 155.1.0.3

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.0.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 113/117/124 ms
```

Rack1R1#ping 155.1.0.5

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.0.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/61 ms
```

Rack1R1#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 155.1.0.5, 100 ms
```

Rack1R3#ping 155.1.0.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 116/117/125 ms

Rack1R3#ping 155.1.0.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.5, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms

Rack1R3#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 155.1.0.5, 117 ms

Rack1R5#ping 155.1.0.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/61 ms

Rack1R5#ping 155.1.0.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 60/60/61 ms

Rack1R5#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 155.1.0.3, 120 ms

Reply to request 0 from 155.1.0.1, 136 ms

2.8 Hub-and-Spoke & Point-to-Point Subinterfaces

- Reset the Frame Relay interfaces of R1, R3, and R5 to its default configuration.
- Enable Frame Relay on the Serial interfaces of R1, R3, and R5 connected to the Frame Relay switch.
- Configure a point-to-point subinterface numbered 105 on R1 and assign it the IP address 155.X.0.1/24 & DLCI 105.
- Configure a point-to-point subinterface numbered 305 on R3 and assign it the IP address 155.X.0.3/24 & DLCI 305.
- Assign the IP address 155.X.0.5 to R5's main Serial interface and configure static IP to DLCI mappings for R1 and R3.
- Ensure that R1 & R5 and R3 & R5 have unicast, multicast, and broadcast reachability to each other, that unicast traffic between R1 & R3 transits through R5.

Configuration

```
R1:
interface Serial0/0
  encapsulation frame-relay
!
interface Serial0/0.105 point-to-point
  ip address 155.1.0.1 255.255.255.0
  frame-relay interface-dlci 105

R3:
interface Serial1/0
  encapsulation frame-relay
!
interface Serial1/0.305 point-to-point
  ip address 155.1.0.3 255.255.255.0
  frame-relay interface-dlci 305

R5:
interface Serial0/0
  encapsulation frame-relay
  ip address 155.1.0.5 255.255.255.0
  frame-relay map ip 155.1.0.1 501 broadcast
  frame-relay map ip 155.1.0.3 503 broadcast
```

Verification

Note

Point-to-point interfaces simplify the previous design by eliminating the need for R1 and R3 to maintain the layer 3 to layer 2 mapping table. Since both R1 and R3 only have one possible layer 2 address to use in the network, 105 on R1 and 305 on R3, point-to-point subinterfaces can be used. However, since R5 has two layer 2 addresses, 501 to R1 and 503 to R3, it must use a multipoint interface and maintain a layer 3 to layer 2 mapping table to differentiate traffic going to R1 vs. R3.

Note that the interface type and resolution method are only locally significant on the routers. This means that R5 does not know that R1 and R3 are using point-to-point subinterfaces from a layer 2 perspective. However other higher layer protocol designs, such as running OSPF over this network, can be affected by the underlying layer 2 network. Strictly from a layer 2 point of view though this scenario has the same exact reachability results as the last two scenarios.

Rack1R1#show frame-relay map

```
Serial0/0.105 (up): point-to-point dlci, dlci 105(0x69,0x1890), broadcast
                    status defined, active
```

Rack1R3#show frame-relay map

```
Serial1/0.305 (up): point-to-point dlci, dlci 305(0x131,0x4C10), broadcast
                    status defined, active
```

Rack1R5#show frame-relay map

```
Serial0/0 (up): ip 155.1.0.1 dlci 501(0x1F5,0x7C50), static,
                broadcast,
                CISCO, status defined, active
Serial0/0 (up): ip 155.1.0.3 dlci 503(0x1F7,0x7C70), static,
                broadcast,
                CISCO, status defined, active
```

Rack1R1#ping 155.1.0.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 112/116/125 ms

Rack1R1#ping 155.1.0.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.5, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

Rack1R1#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 155.1.0.5, 76 ms

Rack1R3#ping 155.1.0.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 112/116/124 ms

Rack1R3#ping 155.1.0.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.5, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/59/64 ms

Rack1R3#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 155.1.0.5, 60 ms

Rack1R5#ping 155.1.0.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

Rack1R5#ping 155.1.0.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 155.1.0.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms

Rack1R5#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 155.1.0.3, 64 ms

Reply to request 0 from 155.1.0.1, 80 ms

2.9 Disabling Inverse-ARP

- Enable Frame Relay on R1, R2, R3, R4, and R5's interfaces connected to the Frame Relay cloud.
- Configure R1's interface with the IP address 155.X.100.1/24.
- Configure R2's interface with the IP address 155.X.100.2/24.
- Configure R3's interface with the IP address 155.X.200.3/24.
- Configure R4's interface with the IP address 155.X.200.4/24.
- Create a multipoint subinterface numbered 100 on R5 with the IP address 155.X.100.5/24 and the DLCIs 501 & 502.
- Create a multipoint subinterface numbered 200 on R5 with the IP address 155.X.200.5/24 and the DLCIs 503 & 504.
- Configure R1 and R2 so that they will not send Inverse-ARP requests for IP out any DLCIs learned from the Frame Relay cloud other than those in the diagram.
- Configure R3 and R4 to disable Inverse-ARP on DLCIs learned from the Frame Relay cloud that are not listed in the diagram by assigning them to the subinterface 999.
- Ensure that R5 has unicast, multicast, and broadcast reachability to R1, R2, R3, and R4.
- R1 and R2 should be able to reach R5, but not each other.
- R3 and R4 should be able to reach R5, but not each other.
- R1, R2, R3, and R4 should not be able to reach each other.

Configuration

```
R1:
interface Serial0/0
 encapsulation frame-relay
 ip address 155.1.100.1 255.255.255.0
 no frame-relay inverse-arp ip 102
 no frame-relay inverse-arp ip 103
 no frame-relay inverse-arp ip 104
 no frame-relay inverse-arp ip 113
```

```
R2:
interface Serial0/0
 encapsulation frame-relay
 ip address 155.1.100.2 255.255.255.0
 no frame-relay inverse-arp ip 201
 no frame-relay inverse-arp ip 203
 no frame-relay inverse-arp ip 204
 no frame-relay inverse-arp ip 213
```

```
R3:
interface Serial1/0
  encapsulation frame-relay
  ip address 155.1.200.3 255.255.255.0
!
interface Serial1/0.999 multipoint
  frame-relay interface-dlci 301
  frame-relay interface-dlci 302
  frame-relay interface-dlci 304
```

```
R4:
interface Serial0/0
  encapsulation frame-relay
  ip address 155.1.200.3 255.255.255.0
!
interface Serial0/0.999 multipoint
  frame-relay interface-dlci 401
  frame-relay interface-dlci 402
  frame-relay interface-dlci 403
  frame-relay interface-dlci 413
```

```
R5:
interface Serial0/0
  encapsulation frame-relay
!
interface Serial0/0.100 multipoint
  ip address 155.1.100.5 255.255.255.0
  frame-relay interface-dlci 501
  frame-relay interface-dlci 502
!
interface Serial0/0.200 multipoint
  ip address 155.1.200.5 255.255.255.0
  frame-relay interface-dlci 503
  frame-relay interface-dlci 504
```

Verification

Note

When a layer 3 protocol such as IPv4 is configured on a multipoint Frame Relay interface, Inverse-ARP requests are automatically sent out all PVCs assigned to that interface that are learned from LMI. In this case there is a full mesh of layer 2 circuits in the underlying Frame Relay cloud, however only a portion of these DLCIs are used in the network topology. If Inverse-ARP were not disabled, i.e. just issuing the **encapsulation frame-relay** and the **ip address** commands, R1, R2, R3, and R4 would have formed a full mesh of Frame Relay mappings, even though they are not all in the same IP subnet. From a layer 2 perspective this may not be an issue, but when higher layer protocols such as EIGRP or Multicast are enabled problems will arise.

In this design R1 and R2 prevent the problem by disabling the Inverse-ARP request on the circuits with the **no frame-relay inverse-arp** command. On R3 and R4 this problem is prevented by moving the DLCIs learned via LMI to a multipoint subinterface that does not have an IP address assigned. Since the subinterface is not running IPv4, an Inverse-ARP request cannot be sent out the PVCs assigned to that interface.

Rack1R1#show frame-relay map

```
Serial0/0 (up): ip 155.1.100.5 dlci 105(0x69,0x1890), dynamic,  
                broadcast,, status defined, active
```

Rack1R2#show frame-relay map

```
Serial0/0 (up): ip 155.1.100.5 dlci 205(0xCD,0x30D0), dynamic,  
                broadcast,, status defined, active
```

Rack1R3#show frame-relay map

```
Serial1/0 (up): ip 155.1.200.5 dlci 305(0x131,0x4C10), dynamic,  
                broadcast,, status defined, active
```

Rack1R4#show frame-relay map

```
Serial0/0 (up): ip 155.1.200.5 dlci 405(0x195,0x6450), dynamic,  
                broadcast,  
                CISCO, status defined, active
```

Rack1R5#show frame-relay map

```
Serial0/0.100 (up): ip 155.1.100.1 dlci 501(0x1F5,0x7C50), dynamic,
                    broadcast,, status defined, active
Serial0/0.100 (up): ip 155.1.100.2 dlci 502(0x1F6,0x7C60), dynamic,
                    broadcast,, status defined, active
Serial0/0.200 (up): ip 155.1.200.3 dlci 503(0x1F7,0x7C70), dynamic,
                    broadcast,
                    CISCO, status defined, active
Serial0/0.200 (up): ip 155.1.200.4 dlci 504(0x1F8,0x7C80), dynamic,
                    broadcast,
                    CISCO, status defined, active
```

Rack1R5#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 155.1.100.2, 96 ms
Reply to request 0 from 155.1.200.3, 196 ms
Reply to request 0 from 155.1.200.4, 124 ms
Reply to request 0 from 155.1.100.1, 112 ms
```

2.10 Back-to-Back Frame Relay

- Configure IP addresses on the directly connected Serial link between R4 and R5 using the information in the diagram.
- Configure Frame Relay PVC 514 on this link without using LMI.
- Configure static IP to DLCI mappings on R4 and R5 to obtain reachability on this segment.

Configuration

```
R4:
interface Serial0/1
 ip address 155.1.45.4 255.255.255.0
 encapsulation frame-relay
 no keepalive
 frame-relay map ip 155.1.45.5 514 broadcast
```

```
R5:
interface Serial0/1
 ip address 155.1.45.5 255.255.255.0
 encapsulation frame-relay
 no keepalive
 clock rate 64000
 frame-relay map ip 155.1.45.4 514 broadcast
```

Verification

Note

Back-to-Back Frame Relay can be used between routers connected directly by a Serial link. Since there is no Frame Relay service provider that is advertising PVCs through LMI, the **no keepalive** command is used to disable LMI. The result of this is seen in the **show frame-relay pvc** output with the PVC status of **STATIC**. **STATIC** means that the router is not receiving LMI updates about the circuit, and it can never move to the **INACTIVE** state due to an LMI message.

```
Rack1R4#show frame-relay pvc 514
```

```
PVC Statistics for interface Serial0/1 (Frame Relay DTE)
```

```
DLCI = 514, DLCI USAGE = LOCAL, PVC STATUS = STATIC, INTERFACE = Serial0/1
```

```
input pkts 66          output pkts 66          in bytes 6864
out bytes 6864        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 0      out bcast bytes 0
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:01:52, last time pvc status changed 00:01:52
```

Rack1R4#show frame-relay map

```
Serial0/0 (up): ip 155.1.200.5 dlci 405(0x195,0x6450), dynamic,
                broadcast,
                CISCO, status defined, active
Serial0/1 (up): ip 155.1.45.5 dlci 514(0x202,0x8020), static,
                broadcast,
                CISCO
```

Rack1R4#ping 155.1.45.5

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.45.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
```

Rack1R4#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 155.1.45.5, 32 ms
Reply to request 0 from 155.1.200.5, 60 ms
```

Rack1R5#show frame-relay pvc 514

PVC Statistics for interface Serial0/1 (Frame Relay DTE)

DLCI = 514, DLCI USAGE = LOCAL, PVC STATUS = **STATIC**, INTERFACE = Serial0/1

```
input pkts 73          output pkts 101          in bytes 7592
out bytes 10504        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 0      out bcast bytes 0
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:05:40, last time pvc status changed 00:05:30
```

Rack1R5#show frame-relay map

```
Serial0/0.100 (up): ip 155.1.100.1 dlci 501(0x1F5,0x7C50), dynamic,
                broadcast,, status defined, active
Serial0/0.100 (up): ip 155.1.100.2 dlci 502(0x1F6,0x7C60), dynamic,
                broadcast,, status defined, active
Serial0/0.200 (up): ip 155.1.200.3 dlci 503(0x1F7,0x7C70), dynamic,
                broadcast,
                CISCO, status defined, active
Serial0/0.200 (up): ip 155.1.200.4 dlci 504(0x1F8,0x7C80), dynamic,
                broadcast,
                CISCO, status defined, active
Serial0/1 (up): ip 155.1.45.4 dlci 514(0x202,0x8020), static,
                broadcast,
                CISCO
```

2.11 Frame Relay End-to-End Keepalives

- Create a Frame Relay map-class called FREEK on R4 and R5.
- Configure R4 to respond to Frame Relay keepalives, and R5 to request them.
- Set R5 to poll every 5 seconds with an event window of 10 and an error threshold of 8.
- R4 should apply this configuration to its main Serial0/0 interface, while R5 should apply this just to the DLCI connecting to R4.

Configuration

R4:

```
map-class frame-relay FREEK
  frame-relay end-to-end keepalive mode reply
  frame-relay end-to-end keepalive timer recv 5
  frame-relay end-to-end keepalive event-window recv 10
  frame-relay end-to-end keepalive error-threshold recv 8
!
interface Serial0/0
  frame-relay class FREEK
```

R5:

```
map-class frame-relay FREEK
  frame-relay end-to-end keepalive mode request
  frame-relay end-to-end keepalive timer send 5
  frame-relay end-to-end keepalive event-window send 10
  frame-relay end-to-end keepalive error-threshold send 8
!
interface Serial0/0.200 multipoint
  frame-relay interface-dlci 504
  class FREEK
```


Verification **Note**

In true end-to-end Frame Relay networks the LMI status that one router receives indicates the end-to-end status of the circuit that the other router receives. This consists of the status from the router to the switch via the User-to-Network interface (UNI), the status of all the switch to switch links via the Network-to-Network (NNI) interface, and the remote UNI. However in the majority of real world service provider networks today Frame Relay only exists on the edge, and is transported in a carrier protocol such as MPLS in the transit. The result of this is that the LMI status of one end of the link does not update the LMI status of the remote end.

The Frame Relay End-to-End Keepalive (EEK) feature avoids design problems like this by adding an additional keepalive layer that is exchanged directly between the router endpoints. When a router is configured to “send” EEK, it implies that it is waiting to “receive” EEK. Likewise if a router is configured to “receive” EEK it is expecting the “send” to come in. This means that if the receive side does not hear the request come in after the event timers expire, or the send side does not hear the reply come in, the PVC status changes to EEK DOWN, and traffic cannot be sent out the link.

```
Rack1R4#show frame-relay end-to-end keepalive interface serial0/0 405
```

```
End-to-end Keepalive Statistics for interface Serial0/0 (Frame Relay DTE)
```

```
DLCI = 405, DLCI USAGE = LOCAL, VC STATUS = ACTIVE (EEK UP)
```

```
RECEIVE SIDE STATISTICS
```

```
Send Sequence Number: 18,          Receive Sequence Number: 17
Configured Event Window: 10,      Configured Error Threshold: 8
Total Observed Events: 30,        Total Observed Errors: 10
Monitored Events: 10,             Monitored Errors: 0
Successive Successes: 10,         End-to-end VC Status: UP
```

```
Rack1R5#show frame-relay end-to-end keepalive interface serial0/0
```

```
End-to-end Keepalive Statistics for Interface Serial0/0 (Frame Relay DTE)
```

```
DLCI = 504, DLCI USAGE = LOCAL, VC STATUS = ACTIVE (EEK UP)
```

```
SEND SIDE STATISTICS
```

```
Send Sequence Number: 22,          Receive Sequence Number: 23
Configured Event Window: 10,      Configured Error Threshold: 8
Total Observed Events: 25,        Total Observed Errors: 0
Monitored Events: 10,             Monitored Errors: 0
Successive Successes: 10,         End-to-end VC Status: UP
```

2.12 Frame Relay Broadcast Queue

- Tune the Frame Relay broadcast queue size between R4 and R5 to 100 packets.

Configuration

```
R4:  
interface Serial0/0  
  frame-relay broadcast-queue 100 256000 36
```

```
R5:  
interface Serial0/0  
  frame-relay broadcast-queue 100 256000 36
```

2.13 Frame Relay TCP & RTP Header Compression

- Enable RTP header compression for DLCIs 503 and 305 on R5 and R3 respectively.
- Enable TCP header compression for DLCIs 501 and 105 on R5 and R1 respectively.

Configuration

```
R1:
interface Serial0/0
  frame-relay map ip 155.1.100.5 105 broadcast tcp header-compression

R3:
interface Serial1/0
  frame-relay map ip 155.1.200.5 305 broadcast rtp header-compression

R5:
interface Serial0/0.100 multipoint
  frame-relay map ip 155.1.100.1 501 broadcast tcp header-compression
!
interface Serial0/0.200 multipoint
  frame-relay map ip 155.1.200.3 503 broadcast rtp header-compression
```

Verification

Rack1R5#show frame-relay ip tcp header-compression

```
DLCI 501      Link/Destination info: ip 155.1.100.1
Interface Serial0/0.100 DLCI 501 (compression on, VJ)
  Rcvd:      67 total, 65 compressed, 0 errors, 0 status msgs
             0 dropped, 0 buffer copies, 0 buffer failures
  Sent:      46 total, 43 compressed, 0 status msgs, 0 not predicted
             1473 bytes saved, 435 bytes sent
             4.38 efficiency improvement factor
  Connect: 256 rx slots, 256 tx slots,
             2 misses, 0 collisions, 0 negative cache hits, 255 free contexts
             95% hit ratio, five minute miss rate 0 misses/sec, 0 max
```

Rack1R5#show frame-relay ip rtp header-compression

```
DLCI 503      Link/Destination info: ip 155.1.200.3
Interface Serial0/0.200 DLCI 503 (compression on, Cisco, RTP)
  Rcvd:      0 total, 0 compressed, 0 errors, 0 status msgs
             0 dropped, 0 buffer copies, 0 buffer failures
  Sent:      0 total, 0 compressed, 0 status msgs, 0 not predicted
             0 bytes saved, 0 bytes sent
  Connect: 256 rx slots, 256 tx slots,
             0 misses, 0 collisions, 0 negative cache hits, 256 free contexts
```

2.13 PPP over Frame Relay

- Create Virtual-Template interfaces on R1 and R5 using the IP addresses 155.X.0.1/24 and 155.X.0.5/24 respectively.
- Enable Frame Relay on R1 and R5's Serial interfaces attached to the Frame Relay cloud, and attach the Virtual-Templates to the DLCIs connecting these devices.
- Ensure that R1 and R5 have unicast, multicast, and broadcast reachability to each other over this segment.

Configuration

```
R1:
interface Serial0/0
  encapsulation frame-relay
  frame-relay interface-dlci 105 ppp Virtual-Template1
!
interface Virtual-Template1
  ip address 155.1.0.1 255.255.255.0
```

```
R5:
interface Serial0/0
  encapsulation frame-relay
  frame-relay interface-dlci 501 ppp Virtual-Template1
!
interface Virtual-Template1
  ip address 155.1.0.5 255.255.255.0
```

Verification

Note

Running PPP over Frame Relay (PPPoFR) adds additional features such as authentication that are not available in the Frame Relay header itself. Most issues that relate to PPPoFR come from the order of operations. The first step in this configuration should be to configure the Virtual-Template interface. On this interface all logical options such as PPP authentication or IP addressing should be configured. Next the Virtual-Template should be bound to the Frame Relay DLCI with the **frame-relay interface-dlci xxx ppp Virtual-Template** command. If bound properly a log message should appear that the *Virtual-Access* interface's line protocol changed state to up. This means that the logical instance of the PPP link is actually active. When viewing the **show ip interface brief** or **show ip route** output, this is the interface that should be in the UP/UP state.

Rack1R1#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	manual	administratively down	down
Serial0/0	unassigned	YES	TFTP	up	up
Serial0/0.105	unassigned	YES	manual	deleted	down
Serial0/1	unassigned	YES	manual	administratively down	down
NV10	unassigned	NO	unset	up	up
Virtual-Access1	unassigned	YES	unset	down	down
Virtual-Template1	155.1.0.1	YES	manual	down	down
Virtual-Access2	155.1.0.1	YES	TFTP	up	up

Rack1R1#show interface virtual-access2

```
Virtual-Access2 is up, line protocol is up
  Hardware is Virtual Access interface
  Internet address is 155.1.0.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open
  Open: IPCP
  PPPoFR vaccess, cloned from Virtual-Template1
  Vaccess status 0x44
  Bound to Serial0/0 DLCI 105, Cloned from Virtual-Template1, loopback not set
  Keepalive set (10 sec)
  DTR is pulsed for 5 seconds on reset
  Last input 00:02:00, output never, output hang never
  Last clearing of "show interface" counters 00:02:17
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
<output omitted>
```

By default PPP uses the peer-neighbor route in order to solve any routing problems due to unnumbered links or different IP address pools in DHCP. If the PPP link is up and functional the peer-neighbor route, which is the remote end's IP address, should appear in the routing table as a connected route.

```
Rack1R1#show ip route connected
      155.1.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       155.1.0.0/24 is directly connected, Virtual-Access2
C       155.1.0.5/32 is directly connected, Virtual-Access2
```

Since PPP is encapsulated between the Frame Relay and the normal layer 3 header, the Frame Relay process does not require layer 3 to layer 2 resolution for any upper layer protocols that are configured on top of PPP. This means that both unicast and multicast/broadcast transmission are successful for IPv4 without the need for a **frame-relay map** command, even though this configuration uses the main Serial interfaces, which are usually multipoint. This is due to the fact that the Virtual-Access is a point-to-point link, since by definition PPP is a point-to-point protocol.

```
Rack1R1#ping 155.1.0.5
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.0.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/61 ms
```

```
Rack1R1#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 155.1.0.5, 64 ms
```

2.14 Bridging over Frame Relay

- Remove the previous PPPoFR configuration on R5.
- Configure ports Fa0/2 and Fa0/24 on SW2 in VLAN 100.
- Configure SW1's interface Fa0/5 with the IP address 192.10.X.7/24.
- Enable Frame Relay on the Serial interfaces connecting to the Frame Relay cloud on R2 and R5.
- Disable IP routing on R2 and R5 and remove any IP addresses from their interfaces.
- Configure an IEEE STP bridge group numbered 1 on R2 & R5, and apply this to their Frame Relay connections and their FastEthernet0/0 interfaces.
- Ensure that SW1 and BB2 have unicast, multicast, and broadcast reachability to each other.

Configuration

```
R2:
interface FastEthernet0/0
  bridge-group 1
!
interface Serial0/0
  encapsulation frame-relay
  frame-relay map bridge 205 broadcast
  bridge-group 1
!
bridge 1 protocol ieee

R5:
interface FastEthernet0/0
  bridge-group 1
!
interface Serial0/0
  encapsulation frame-relay
  frame-relay map bridge 502 broadcast
  bridge-group 1
!
bridge 1 protocol ieee

SW1:
interface FastEthernet0/5
  no switchport
  ip address 192.10.1.7 255.255.255.0

SW2:
interface FastEthernet0/2
  switchport access vlan 100
!
interface FastEthernet0/24
  switchport access vlan 100
```

Verification

Note

When bridging over multipoint Frame Relay interfaces the **frame-relay map bridge** command is required to tell the router which PVC to encapsulate Spanning-Tree Protocol and the other bridged packets on the interface.

Once completed the below three verifications all imply that the bridge domain is built properly. From the **show spanning-tree** output on R2 and R5 we can see that they both agree on who the root bridge is. This implies that they see each other's STP BPDUs, and are in the same broadcast domain.

Rack1R2#show spanning-tree 1

```
Bridge group 1 is executing the ieee compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 0012.0120.3680
Configured hello time 2, max age 20, forward delay 15
Current root has priority 32768, address 0012.001c.daa0
Root port is 5 (Serial0/0), cost of root path is 647
Topology change flag not set, detected flag not set
Number of topology changes 7 last change occurred 00:01:15 ago
    from Serial0/0
Times: hold 1, topology change 35, notification 2
    hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0, aging 300
```

```
Port 4 (FastEthernet0/0) of Bridge group 1 is forwarding
Port path cost 19, Port priority 128, Port Identifier 128.4.
Designated root has priority 32768, address 0012.001c.daa0
Designated bridge has priority 32768, address 0012.0120.3680
Designated port id is 128.4, designated path cost 647
Timers: message age 0, forward delay 0, hold 0
Number of transitions to forwarding state: 1
BPDU: sent 357, received 5
```

```
Port 5 (Serial0/0) of Bridge group 1 is forwarding
Port path cost 647, Port priority 128, Port Identifier 128.5.
Designated root has priority 32768, address 0012.001c.daa0
Designated bridge has priority 32768, address 0012.001c.daa0
Designated port id is 128.6, designated path cost 0
Timers: message age 1, forward delay 0, hold 0
Number of transitions to forwarding state: 1
BPDU: sent 2, received 55
```



```
Rack1R5#show spanning-tree 1
```

```
Bridge group 1 is executing the ieee compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 0012.001c.daa0
Configured hello time 2, max age 20, forward delay 15
We are the root of the spanning tree
Topology change flag not set, detected flag not set
Number of topology changes 4 last change occurred 00:01:58 ago
    from Serial0/0
Times: hold 1, topology change 35, notification 2
    hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0, aging 300
```

```
Port 4 (FastEthernet0/0) of Bridge group 1 is forwarding
```

```
Port path cost 19, Port priority 128, Port Identifier 128.4.
Designated root has priority 32768, address 0012.001c.daa0
Designated bridge has priority 32768, address 0012.001c.daa0
Designated port id is 128.4, designated path cost 0
Timers: message age 0, forward delay 0, hold 0
Number of transitions to forwarding state: 1
BPDU: sent 364, received 3
```

```
Port 6 (Serial0/0) of Bridge group 1 is forwarding
```

```
Port path cost 647, Port priority 128, Port Identifier 128.6.
Designated root has priority 32768, address 0012.001c.daa0
Designated bridge has priority 32768, address 0012.001c.daa0
Designated port id is 128.6, designated path cost 0
Timers: message age 0, forward delay 0, hold 0
Number of transitions to forwarding state: 1
BPDU: sent 368, received 8
```

A traceroute on SW1 shows only one hop to the destination. This implies that there are no layer 3 routing devices in the transit path, only layer 2 bridges.

```
Rack1SW1#traceroute 192.10.1.254
```

```
Type escape sequence to abort.
Tracing the route to 192.10.1.254
```

```
 1 192.10.1.254 42 msec * 42 msec
```

By pinging the all subnet broadcast address all devices that are directly connected in the broadcast domain should respond. Since SW1 gets a response from BB2, via 192.10.1.254, this implies that SW1's interface 192.10.1.7 is in the same broadcast domain as BB2.

```
Rack1SW1#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 192.10.1.254, 67 ms
```

