



## Configuring Unidirectional Link Routing

---

This chapter describes the unidirectional link routing (UDLR) feature. UDLR provides mechanisms for a router to emulate a bidirectional link to enable the routing of unicast and multicast packets over a physical unidirectional interface, such as a broadcast satellite link. However, there must be a back channel or other path between the routers that share a physical unidirectional link (UDL). A UDLR tunnel is a mechanism for unicast and multicast traffic; Internet Group Management Protocol (IGMP) UDLR and IGMP Proxy are mechanisms for multicast traffic.

For information about tunnel interfaces, refer to the “Configuring Logical Interfaces” chapter in the *Cisco IOS Interface Configuration Guide*. For information about IGMP, refer to the chapter “Configuring IP Multicast Routing” in the *Cisco IOS IP Configuration Guide*.

For a complete description of the UDLR commands used in this chapter, refer to the “Unidirectional Link Routing Commands” chapter in the *Cisco IOS IP Command Reference, Volume 3 of 3: Multicast*. To locate documentation of other commands that appear in this chapter, use the command reference master index, or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the “Identifying Supported Platforms” section in the “Using Cisco IOS Software” chapter.

### UDLR Overview

Both unicast and multicast routing protocols forward data on interfaces from which they have received routing control information. This model works only on bidirectional links for most existing routing protocols. However, some networks use broadcast satellite links, which are unidirectional. For networks that use broadcast satellite links, accomplishing two-way communication over broadcast satellite links presents a problem in terms of discovering and sharing knowledge of a network topology.

Specifically, in unicast routing, when a router receives an update message on an interface for a prefix, it forwards data for destinations that match that prefix out that same interface. This is the case in distance vector routing protocols. Similarly, in multicast routing, when a router receives a join message for a multicast group on an interface, it forwards copies of data destined for that group out that same interface. Based on these principles, existing unicast and multicast routing protocols cannot be supported over UDLs. UDLR is designed to enable the operation of routing protocols over UDLs without changing the routing protocols themselves.

UDLR enables a router to emulate the behavior of a bidirectional link for IP operations over UDLs. UDLR has three complementary mechanisms for bidirectional link emulation, which are described in the following sections:

- UDLR Tunnel
- IGMP UDLR
- IGMP Proxy

You can use each mechanism independently or in conjunction with the others.

## UDLR Tunnel

The UDLR tunnel mechanism enables IP and its associated unicast and multicast routing protocols to treat the UDL as being logically bidirectional. A packet that is destined on a receive-only interface is picked up by the UDLR tunnel mechanism and sent to an upstream router using a generic routing encapsulation (GRE) tunnel. The control traffic flows in the opposite direction as the user data flow. When the upstream router receives this packet, the UDLR tunnel mechanism makes it appear that the packet was received on a send-only interface on the UDL.

The purpose of the unidirectional GRE tunnel is to move control packets from a downstream node to an upstream node. The one-way tunnel is mapped to a one-way interface (that goes in the opposite direction). Mapping is performed at the link layer, so the one-way interface appears bidirectional. When the upstream node receives packets over the tunnel, it must make the upper-layer protocols act as if the packets were received on the send-capable UDL.

UDLR tunnel supports the following features:

- Address Resolution Protocol (ARP) and Next Hop Resolution Protocol (NHRP) over a UDL
- Emulation of bidirectional links for all IP traffic (as opposed to only control-only broadcast/multicast traffic)
- Support for IP GRE multipoint at a receive-only tunnel



### Note

A UDL router can have many routing peers, for example, routers interconnected via a broadcast satellite link. As with bidirectional links, the number of peer routers a router has must be kept relatively small to limit the volume of routing updates that must be processed. For multicast operation, we recommend using the IGMP UDLR mechanism when interconnecting more than 20 routers.

## IGMP UDLR

Another mechanism that enables support of multicast routing protocols over UDLs is using IP multicast routing with IGMP, which has been enhanced to accommodate UDLR. This mechanism scales well for many broadcast satellite links.

With IGMP UDLR, an upstream router sends periodic queries for members on the UDL. The queries include a unicast address of the router that is not the unicast address of the unidirectional interface. The downstream routers forward IGMP reports received from directly connected members (on interfaces configured to helper forward IGMP reports) to the upstream router. The upstream router adds the unidirectional interface to the (\*, G) outgoing interface list, thereby enabling multicast packets to be forwarded down the UDL.

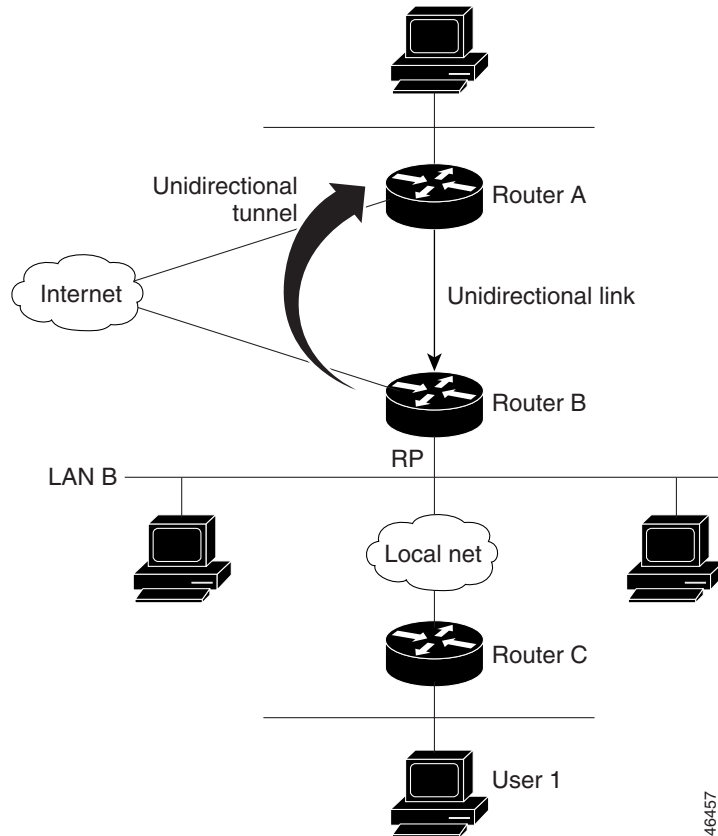
In a large enterprise network, it is not possible to be able to receive IP multicast traffic via satellite and forward the traffic throughout the network. This limitation exists because receiving hosts must be directly connected to the downstream router. However, you can use the IGMP Proxy mechanism to overcome this limitation. See the “IGMP Proxy” section later in this chapter for more information on this mechanism.

For information on IGMP, refer to the “Configuring IP Multicast Routing” chapter in the *Cisco IOS IP Configuration Guide*.

## IGMP Proxy

The IGMP Proxy mechanism enables hosts that are not directly connected to a downstream router to join a multicast group sourced from an upstream network. Figure 82 illustrates this mechanism.

**Figure 82 IGMP Mroute Proxy Mechanism**



46457

In the scenario in [Figure 82](#), the following sequence of events occurs:

1. User 1 joins multicast group G.
2. Router C sends a Protocol Independent Multicast (PIM) join message hop-by-hop to the rendezvous point (Router B).
3. Router B receives the PIM join message and adds a forwarding entry for group G on LAN B.
4. Router B periodically checks its mroute table, and forwards an IGMP report for each multicast group in which it is the reporter.
5. Router A creates and maintains a forwarding entry on the UDL.

In an enterprise network, for example, it is desirable to be able to receive IP multicast traffic via satellite and forward the traffic throughout the network. With IGMP UDLR alone, this scenario is not possible because receiving hosts must be directly connected to the downstream router. IGMP Proxy overcomes this limitation by creating an IGMP report for (\*, G) entries in the multicast forwarding table. To make this scenario functional, you must configure PIM sparse mode (PIM-SM) in the network, make the UDL downstream router the rendezvous point (RP) for a select set of addresses, and configure mroute proxy on interfaces leading to PIM-enabled networks with potential members. When the UDL downstream router has a (\*, G) forwarding entry for an mroute proxy interface, an IGMP report for the group is created and sent to a loopback interface (IGMP Proxy interface). The loopback interface then uses the same mechanism as IGMP UDLR to forward reports upstream.


**Note**


---

Because PIM messages are not forwarded upstream, each downstream network and the upstream network has a separate domain.

---

## UDLR Tunnel Configuration Task List

To configure a UDLR tunnel, perform the required tasks described in the following section:

- [Configuring UDLR Tunnel](#) (Required)

## Prerequisite

Before configuring UDLR tunnel, ensure that all routers on the UDL have the same subnet address. If all routers on the UDL cannot have the same subnet address, the upstream router must be configured with secondary addresses to match all the subnets that the downstream routers are attached to.

## Configuring UDLR Tunnel

When configuring a UDLR tunnel, you must configure both the upstream and downstream routers to meet the following conditions:

- You need not assign an IP address to the tunnel (you need not use the **ip address** or **ip unnumbered** interface configuration commands).
- You must configure the tunnel endpoint addresses.
- The tunnel mode defaults to GRE.

- On the upstream router, where the UDL can only send, you must configure the tunnel to receive. When packets are received over the tunnel, the upper-layer protocols treat the packet as though it is received over the unidirectional, send-only interface.
- On the downstream router, where the UDL can only receive, you must configure the tunnel to send. When packets are sent by upper-layer protocols over the interface, they will be redirected and sent over this GRE tunnel.

To configure a UDLR tunnel on the upstream router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>interface</b> <i>type number</i>	Configures the unidirectional send-only interface.
Step 2	Router(config-if)# <b>interface tunnel</b> <i>number</i>	Configures the receive-only tunnel interface.
Step 3	Router(config-if)# <b>tunnel udlr receive-only</b> <i>type number</i>	Configures the UDLR tunnel. Use the same <i>type</i> and <i>number</i> values as the unidirectional send-only interface <i>type</i> and <i>number</i> values specified with the <b>interface type number</b> command.
Step 4	Router(config-if)# <b>tunnel source</b> { <i>ip-address</i>   <i>type number</i> }	Configures the tunnel source.
Step 5	Router(config-if)# <b>tunnel destination</b> { <i>hostname</i>   <i>ip-address</i> }	Configures the tunnel destination.

To configure a UDLR tunnel on the downstream router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>interface</b> <i>type number</i>	Configures the unidirectional receive-only interface.
Step 2	Router(config-if)# <b>interface tunnel</b> <i>number</i>	Configures the send-only tunnel interface.
Step 3	Router(config-if)# <b>tunnel udlr send-only</b> <i>type number</i>	Configures the UDLR tunnel. Use the same <i>type</i> and <i>number</i> values as the unidirectional receive-only interface <i>type</i> and <i>number</i> values specified with the <b>interface type number</b> command.
Step 4	Router(config-if)# <b>tunnel source</b> { <i>ip-address</i>   <i>type number</i> }	Configures the tunnel source.
Step 5	Router(config-if)# <b>tunnel destination</b> { <i>hostname</i>   <i>ip-address</i> }	Configures the tunnel destination.
Step 6	Router(config-if)# <b>tunnel udlr address-resolution</b>	Enables the forwarding of ARP and NHRP.

See the “UDLR Tunnel Example” section later in this chapter for an example of how to configure a UDLR tunnel. See the “Integrated UDLR Tunnel, IGMP UDLR, and IGMP Proxy Example” section later in this chapter for an example of how to set up all three UDLR mechanisms in the same configuration.

# IGMP UDRL Configuration Task List

To configure IGMP UDRL, perform the tasks described in the following sections. The tasks in the first section are required; the tasks in the remaining sections are optional.

- [Configuring the IGMP UDL](#) (Required)
- [Changing the Distance for the Default RPF Interface](#) (Optional)
- [Monitoring IGMP UDRL](#) (Optional)

## Prerequisites

Before configuring IGMP UDRL, ensure that the following conditions exist:

- All routers on the UDL have the same subnet address. If all routers on the UDL cannot have the same subnet address, the upstream router must be configured with secondary addresses to match all the subnets that the downstream routers are attached to.
- Multicast receivers are directly connected to the downstream routers.

## Configuring the IGMP UDL

To configure an IGMP UDL, you must configure both the upstream and downstream routers to meet the following conditions:

- You need not specify whether the direction is sending or receiving; IGMP learns the direction by the nature of the physical connection.
- When the downstream router receives an IGMP report from a host, the router helps the report to the IGMP querier associated with the UDL interface identified in the **ip igmp helper-address** interface configuration command.

To configure the IGMP UDL on the upstream router, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# <b>ip igmp unidirectional-link</b>	Configures IGMP on the interface to be unidirectional.

To configure the IGMP UDL on the downstream router, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	Router(config-if)# <b>ip igmp unidirectional-link</b>	Configures IGMP on the interface to be unidirectional.
Step 2	Router(config-if)# <b>ip igmp helper-address udl type number</b>	Configures the interface to be an IGMP helper. Use this command on every downstream router, on every interface to a potential multicast receiver. Specify the <i>type</i> and <i>number</i> values that identify the UDL interface.

See the “[IGMP UDLR Example](#)” section later in this chapter for an example of how to configure IGMP UDLR. See the “[Integrated UDLR Tunnel, IGMP UDLR, and IGMP Proxy Example](#)” section later in this chapter for an example of how to set up all three UDLR mechanisms in the same configuration.

## Changing the Distance for the Default RPF Interface

By default, the distance for the default Reverse Path Forwarding (RPF) interface is 15. Any explicit sources learned by routing protocols will take preference if their distance is less than the distance configured by the **ip multicast default-rpf-distance** global configuration command.

If you want IGMP to prefer the UDLR link, set the distance to be less than the distances of the unicast routing protocols. If you want IGMP to prefer the non-UDLR link, set the distance to be greater than the distances of the unicast routing protocols. This task might be required on downstream routers if you want some sources to use RPF to reach the UDLR link and others to use the terrestrial paths.

To change the distance for the default RPF interface, use the following command in global configuration mode:

Command	Purpose
Router(config)# <b>ip multicast default-rpf-distance</b> <i>distance</i>	Changes the distance for the default RPF interface.

## Monitoring IGMP UDLR

To display UDLR information for directly connected groups on interfaces that have a UDL helper address configured, use the following command in EXEC mode:

Command	Purpose
Router# <b>show ip igmp udlr</b> [ <i>group-name</i>   <i>group-address</i>   <i>type number</i> ]	Displays UDLR information for directly connected multicast groups on interfaces that have a UDL helper address configured.

## IGMP Proxy Configuration Task List

To configure IGMP Proxy, perform the tasks described in the following sections. The tasks in the first section are required; the tasks in the remaining section are optional.

- [Configuring IGMP Proxy](#) (Required)
- [Verifying IGMP Proxy](#) (Optional)

## Prerequisites

Before configuring IGMP Proxy, ensure that the following conditions exist:

- All routers on the UDL have the same subnet address. If all routers on the UDL cannot have the same subnet address; the upstream router must be configured with secondary addresses to match all the subnets that the downstream routers are attached to.
- PIM-SM is configured in the network, the UDL downstream router is the RP for a select set of addresses, and mroute proxy is configured on interfaces leading to PIM-enabled networks with potential members.

## Configuring IGMP Proxy

To configure IGMP Proxy, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	Router(config-if)# <b>ip igmp mroute-proxy</b> <i>type number</i>	When used with the <b>ip igmp proxy-service</b> command, enables forwarding of IGMP reports to a proxy service interface for all (*, G) forwarding entries for this interface in the multicast forwarding table.
Step 2	Router(config-if)# <b>ip igmp proxy-service</b>	Enables the mroute proxy service. Based on the IGMP query interval, the router periodically checks the mroute table for (*, G) forwarding entries that match interfaces configured with the <b>ip igmp mroute-proxy</b> command. Where there is a match, one IGMP report is created and received on this interface. This command was intended to be used with the <b>ip igmp helper-address udl</b> command, in which case the IGMP report would be forwarded to an upstream router.

See the “[IGMP Proxy Example](#)” section later in this chapter for an example of how to configure IGMP Proxy. See the “[Integrated UDLR Tunnel, IGMP UDLR, and IGMP Proxy Example](#)” section later in this chapter for an example of how to set up all three UDLR mechanisms in the same configuration.

## Verifying IGMP Proxy

To verify that IGMP Proxy is configured properly, use the **show ip igmp interface EXEC** command. The following sample output shows that IGMP Proxy is configured on Ethernet interface 1/0/6.

```
router# show ip igmp udlr

IGMP UDLR Status, UDL Interfaces:Ethernet1/0/6
Group Address      Interface          UDL Reporter      Reporter Expires
239.1.1.2          Ethernet1/0/6     10.10.0.3         00:02:59
239.1.1.1          Ethernet1/0/6     10.10.0.2         00:02:40
```

# UDLR Configuration Examples

This section provides the following UDLR examples:

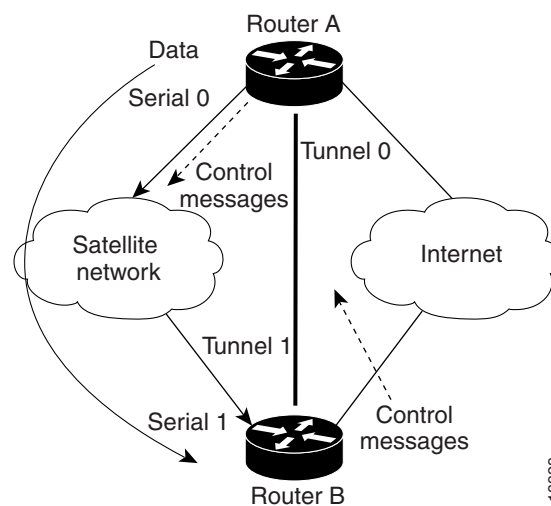
- [UDLR Tunnel Example](#)
- [IGMP UDLR Example](#)
- [IGMP Proxy Example](#)
- [Integrated UDLR Tunnel, IGMP UDLR, and IGMP Proxy Example](#)

## UDLR Tunnel Example

The following example shows how to configure a UDLR tunnel. In the example, Router A (the upstream router) is configured with Open Shortest Path First (OSPF) and PIM. Serial interface 0 has send-only capability. Therefore, the UDLR tunnel is configured as receive only, and points to serial 0.

Router B (the downstream router) is configured with OSPF and PIM. Serial interface 1 has receive-only capability. Therefore, the UDLR tunnel is configured as send-only, and points to serial 1. The forwarding of ARP and NHRP is enabled. [Figure 83](#) illustrates the example.

**Figure 83 UDLR Tunnel Example**



### Router A Configuration

```
ip multicast-routing
!
! Serial0 has send-only capability
!
interface serial 0
 encapsulation hdlc
 ip address 10.1.0.1 255.255.0.0
 ip pim sparse-dense-mode
!
! Configure tunnel as receive-only UDLR tunnel.
!
interface tunnel 0
 tunnel source 11.0.0.1
 tunnel destination 11.0.0.2
```

```

tunnel udlr receive-only serial 0
!
! Configure OSPF.
!
router ospf <pid>
 network 10.0.0.0 0.255.255.255 area 0

```

### Router B Configuration

```

ip multicast-routing
!
! Serial1 has receive-only capability
!
interface serial 1
 encapsulation hdlc
 ip address 10.1.0.2 255.255.0.0
 ip pim sparse-dense-mode

!
! Configure tunnel as send-only UDLR tunnel.
!
interface tunnel 0
 tunnel source 11.0.0.2
 tunnel destination 11.0.0.1
 tunnel udlr send-only serial 1
 tunnel udlr address-resolution
!
! Configure OSPF.
!
router ospf <pid>
 network 10.0.0.0 0.255.255.255 area 0

```

## IGMP UDLR Example

The following example shows how to configure IGMP UDLR. In this example, uplink-rtr is the local upstream router and downlink-rtr is the downstream router. [Figure 84](#) illustrates the example.

Both routers are also connected to each other by a back channel connection. Both routers have two IP addresses: one on the UDL and one on the interface that leads to the back channel. The back channel is any return route and can have any number of routers.



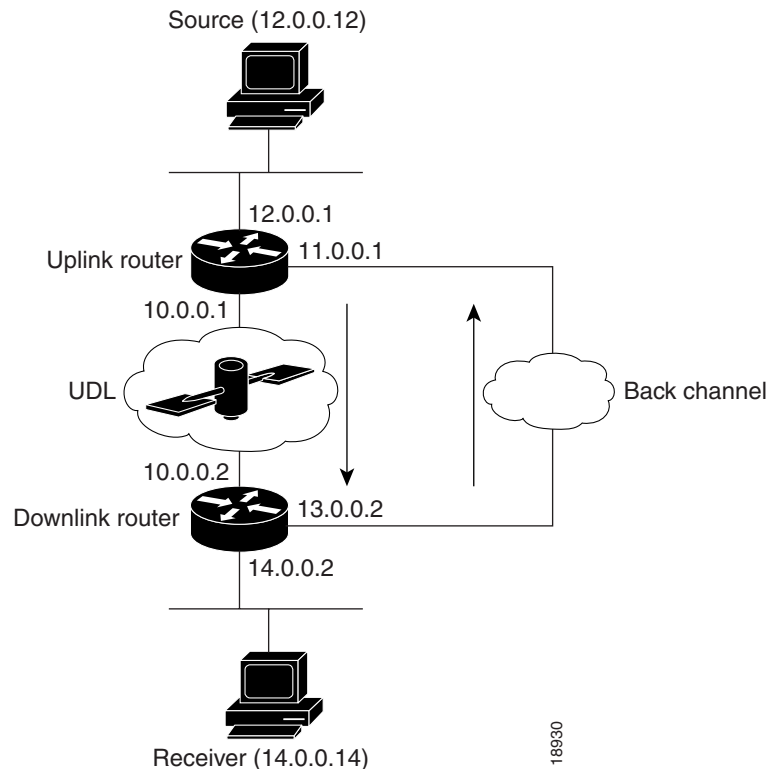
#### Note

---

Configuring PIM on the back channel interfaces on the uplink router and downlink router is optional.

---

All routers on a UDL must have the same subnet address. If all routers on a UDL cannot have the same subnet address, the upstream router must be configured with secondary addresses to match all the subnets that the downstream routers are attached to.

**Figure 84 IGMP Unidirectional Link Routing Example****Uplink Router (uplink-rtr) Configuration**

```

ip multicast-routing
!
! Interface that source is attached to
!
interface ethernet 0
description Typical IP multicast enabled interface
ip address 12.0.0.1 255.0.0.0
ip pim sparse-dense-mode
!
! Back channel
!
interface ethernet 1
description Back channel which has connectivity to downlink-rtr
ip address 11.0.0.1 255.0.0.0
ip pim sparse-dense-mode
!
! Unidirectional link
!
interface serial 0
description Unidirectional to downlink-rtr
ip address 10.0.0.1 255.0.0.0
ip pim sparse-dense-mode
ip igmp unidirectional-link
no keepalive

```

**Downlink Router (downlink-rtr) Configuration**

```

ip multicast-routing
!
! Interface that receiver is attached to, configure for IGMP reports to be

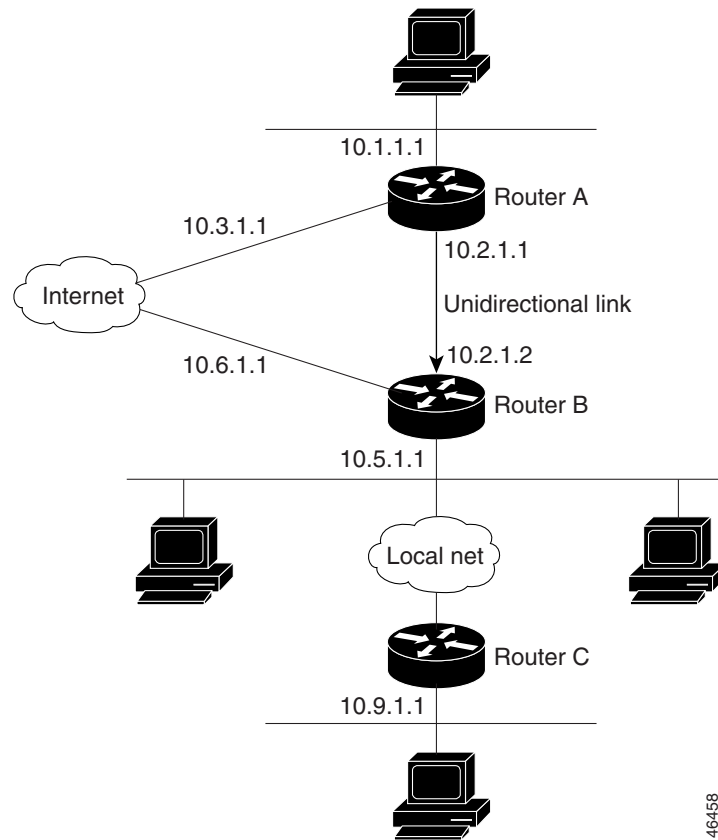
```

```
! helpered for the unidirectional interface.
!
interface ethernet 0
  description Typical IP multicast-enabled interface
  ip address 14.0.0.2 255.0.0.0
  ip pim sparse-dense-mode
  ip igmp helper-address udl serial 0
!
! Back channel
!
interface ethernet 1
  description Back channel that has connectivity to downlink-rtr
  ip address 13.0.0.2 255.0.0.0
  ip pim sparse-dense-mode
!
! Unidirectional link
!
interface serial 0
  description Unidirectional to uplink-rtr
  ip address 10.0.0.2 255.0.0.0
  ip pim sparse-dense-mode
  ip igmp unidirectional-link
  no keepalive
```

## IGMP Proxy Example

The following example shows how to configure IGMP Proxy. In this example, Router C sends a PIM-SM join message to Router B for multicast group G. Router B will send a request to Router A for an IGMP report for group G. Router A will then forward group G multicast traffic over the UDL. [Figure 85](#) illustrates this example.

Figure 85 IGMP Mroute Proxy Topology

**Router A Configuration**

```
interface ethernet 0
ip address 10.1.1.1 255.255.255.0
ip pim dense-mode
!
interface ethernet 1
ip address 10.2.1.1 255.255.255.0
ip pim dense-mode
ip igmp unidirectional link
!
interface ethernet 2
ip address 10.3.1.1 255.255.255.0
```

**Router B Configuration**

```
ip pim rp-address 10.5.1.1 5
access-list 5 permit 239.0.0.0 0.255.255.255.255
!
interface loopback 0
ip address 10.7.1.1 255.255.255.0
ip pim dense-mode
ip igmp helper-address udl ethernet 0
ip igmp proxy-service
!
interface ethernet 0
ip address 10.2.1.2 255.255.255.0
ip pim dense-mode
ip igmp unidirectional link
```

```

!
interface ethernet 1
ip address 10.5.1.1 255.255.255.0
ip pim sparse-mode
ip igmp mroute-proxy loopback 0
!
interface ethernet 2
ip address 10.6.1.1 255.255.255.0

```

### Router C Configuration

```

ip pim rp-address 10.5.1.1 5
access-list 5 permit 239.0.0.0 0.255.255.255
!
interface ethernet 0
ip address 10.8.1.1 255.255.255.0
ip pim sparse-mode
!
interface ethernet 1
ip address 10.9.1.1 255.255.255.0
ip pim sparse-mode

```

## Integrated UDLR Tunnel, IGMP UDLR, and IGMP Proxy Example

The following example shows how to configure UDLR tunnels, IGMP UDLR, and IGMP Proxy on both the upstream and downstream routers sharing a UDL:

### Upstream Configuration

```

ip multicast-routing
!
!
!
interface Tunnel0
ip address 9.1.89.97 255.255.255.252
no ip directed-broadcast
tunnel source 9.1.89.97
tunnel mode gre multipoint
tunnel key 5
tunnel udlr receive-only Ethernet2/3
!
interface Ethernet2/0
no ip address
shutdown
!
! user network
interface Ethernet2/1
ip address 9.1.89.1 255.255.255.240
no ip directed-broadcast
ip pim dense-mode
ip cgmp
fair-queue 64 256 128
no cdp enable
ip rsvp bandwidth 1000 100
!
interface Ethernet2/2
ip address 9.1.95.1 255.255.255.240
no ip directed-broadcast
!
! physical send-only interface
interface Ethernet2/3
ip address 9.1.92.100 255.255.255.240

```

```

no ip directed-broadcast
ip pim dense-mode
ip nhrp network-id 5
ip nhrp server-only
ip igmp unidirectional-link
fair-queue 64 256 31
ip rsvp bandwidth 1000 100
!
router ospf 1
  network 9.1.92.96 0.0.0.15 area 1
!
ip classless
ip route 9.1.90.0 255.255.255.0 9.1.92.99
!

```

### Downstream Configuration

```

ip multicast-routing
!
!
!
interface Loopback0
  ip address 9.1.90.161 255.255.255.252
  ip pim sparse-mode
  ip igmp helper-address udl Ethernet2/3
  ip igmp proxy-service
!
interface Tunnel0
  ip address 9.1.90.97 255.255.255.252
  ip access-group 120 out
  no ip directed-broadcast
  no ip mroute-cache
  tunnel source 9.1.90.97
  tunnel destination 9.1.89.97
  tunnel key 5
  tunnel udlr send-only Ethernet2/3
  tunnel udlr address-resolution
!
interface Ethernet2/0
  no ip address
  no ip directed-broadcast
  shutdown
  no cdp enable
!
! user network
interface Ethernet2/1
  ip address 9.1.90.1 255.255.255.240
  no ip directed-broadcast
  ip pim sparse-mode
  ip igmp mroute-proxy Loopback0
  no cdp enable
!
! Backchannel
interface Ethernet2/2
  ip address 9.1.95.3 255.255.255.240
  no ip directed-broadcast
  no cdp enable
!
! physical receive-only interface
interface Ethernet2/3
  ip address 9.1.92.99 255.255.255.240
  no ip directed-broadcast
  ip pim sparse-mode
  ip igmp unidirectional-link

```

```
no keepalive
no cdp enable
!
router ospf 1
 network 9.1.90.0 0.0.0.255 area 1
 network 9.1.92.96 0.0.0.15 area 1
!
ip classless
ip route 0.0.0.0 0.0.0.0 9.1.95.1
! set rpf to be the physical receive-only interface
ip mroute 0.0.0.0 0.0.0.0 9.1.92.96
ip pim rp-address 9.1.90.1
!
! permit ospf, ping and rsvp, deny others
access-list 120 permit icmp any any
access-list 120 permit 46 any any
access-list 120 permit ospf any any
```