

Advanced Features

This chapter describes advanced features of DLSw+, the benefits they provide, and when and how to use them. Use this chapter to determine which options you want to implement and to learn how to configure those options to address your requirements.

DLSw+ includes features to enhance availability (load balancing, redundancy, and backup peers), improve performance (encapsulation options), minimize broadcasts (ring lists), and build meshed networks (border peers and peer groups). DLSw+ also provides a feature to maximize central site resources and minimize carrier costs (dynamic peers).

Advanced features are optional and do not apply in all networks. Each feature includes a description of where it should be used. Tuning features are covered in the next chapter.

Load Balancing and Redundancy

If you have multiple central site routers supporting DLSw+ for either load balancing or redundancy, read this section. It describes how to balance traffic across multiple central site routers or multiple ports on a single router and how they affect the different phases of operation.

Load balancing in these cases do not refer to balancing traffic across multiple WAN links or IP paths. That load balancing is done by the underlying IP protocol and is transparent to DLSw+.

To understand load balancing, it is useful to understand how DLSw+ peers establish peer connections and find resources. When DLSw+ routers are activated, the first thing they do is establish peer connections with each configured remote peer (unless passive is specified, in which case a peer will wait for the remote peer to initiate a peer connection or unless dynamic is specified and a peer will wait until it has traffic to send). The routers then exchange their capabilities. Included in the capabilities exchange are any resources configured in the global `dlsw icanreach` or `dlsw icannotreach` commands. After the capabilities exchange, the DLSw+ peers are idle until an end system sends an explorer frame (explorer frames are SNA TEST or XID frames or NetBIOS NAME-QUERY or ADD NAME-QUERY frames). Before a cache is populated, explorer frames are forwarded to every active peer and any local ports (other than the port it was received on). It is possible that an end system can be found through multiple remote peers or local ports. The path selected for a given circuit depends on certain advanced configuration options described in this section.

If DLSw+ gets multiple positive replies to an explorer, it will cache up to four peers that can be used to reach a remote end system and up to four ports that can be used to reach a local end system. How these cache entries are used depends on the type of load balancing, if any, is specified.

Fault-Tolerant Mode

If load balancing is not specified, DLSw+ handles multiple paths in fault-tolerant mode. In normal operations, a peer selects the first path in the cache and sets up all circuits via that path unless the path is unavailable. The first path in the cache list can be one of the following:

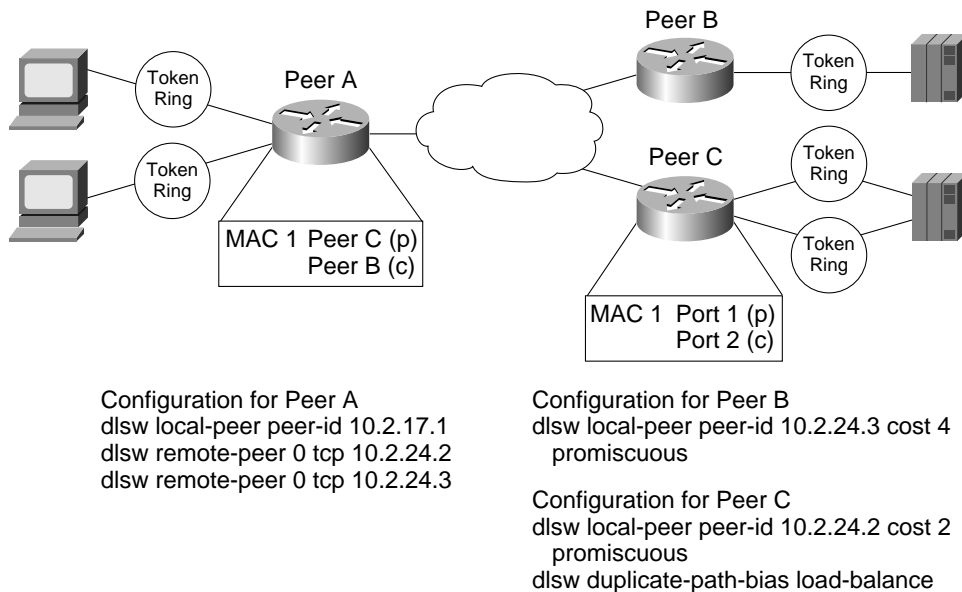
- Peer from which the first positive response was received
- Peer with the least cost
- Port over which the first positive response was received

Configuring Cost

Although configuring cost on a peer is not required, it can be used to control which path the sessions use. Cost can be specified on either a `dlsw local-peer` or a `dlsw remote-peer` command. When specified on a `dlsw local-peer` command, it is exchanged with remote DLSw+ peers as part of the capabilities exchange. The cost configured on the `dlsw remote-peer` command overrides any cost value learned from another devices `dlsw local-peer` command.

In Figure 3-1, there are two channel gateways and three Token Ring adapters that can be used to access mainframe applications. All three adapters have been assigned the same MAC address. Assigning duplicate addresses is a common technique for providing load balancing and redundancy in source-route bridging (SRB) environments. It works because SRB assumes that there are three paths to find the same device and not duplicate LAN addresses. (This technique does not work with transparent bridging [TB].)

Figure 3-1 Possible Configuration and the Resulting Cache Entries Created if All Channel Gateways Illustrated Have the Same MAC Address



In this example, Peer A has `dlsw remote-peer` commands for both Peer B and Peer C. Peer B specifies a cost of 4 in its `dlsw local-peer` command and Peer C specifies a cost of 2. This cost information is exchanged with Peer A during the capabilities exchange.



Note: The output of the `show dlsw capabilities` command displays the cost learned from the other devices rather than what is actually configured on the local peer. For example, the output of the `show dlsw capabilities` command on Peer A will show a cost value of 4 for remote peer B and a cost value of 2 for remote peer C. To determine the cost configured on the local device, issue the `show running configuration` command on the local peer.

When the SNA end system (that is, the PU) on the left sends an explorer packet, Peer A forwards the explorer to both Peer B and Peer C. Peer B and Peer C forward the explorer on their local LAN. Peer B will receive a positive reply to the explorer and send a positive response back to Peer A. Peer C will receive two positive replies (one from each port) and will send a positive reply back to Peer A. Peer C records that it has two ports it can use to reach the MAC address of the channel gateway, and Peer A records that it has two peers it can use to reach the MAC address of the channel gateway.

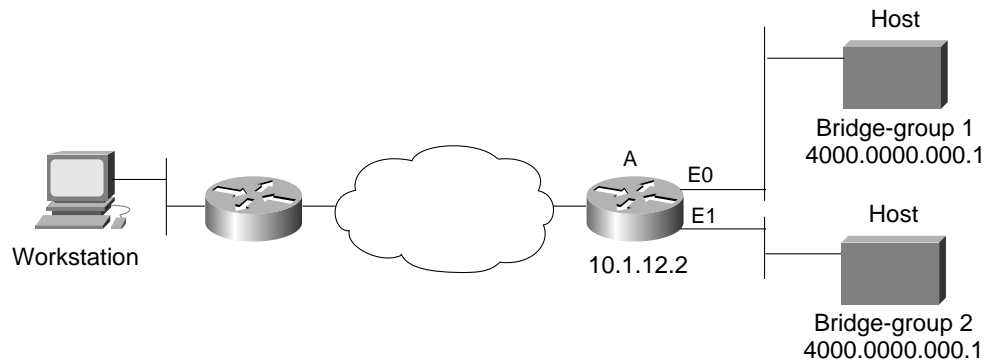
Peer A will forward a positive response to the SNA PU and then establish an end-to-end circuit using Peer C. Peer C is selected because Peer C has a lower cost specified. When the next PU attempts to set up a connection to the same MAC address, it will be set up using Peer C, if available.

At Peer C, the first circuit will be established using Port 1, but the next circuit will use Port 2. This is because Peer C has specified the `round-robin` keyword in the `dlsw load-balance` command. Each new SNA PU will use the next path in the list in a round-robin fashion. See the “Load Balancing Mode” section for more details.

Figure 3-1 shows how to cause all remote connections to prefer one peer over another, but the central site load balances traffic across all the LAN adapters on a given channel gateway. Alternately, load balancing can be specified everywhere to load balance traffic across all central site routers, channel gateways, and LANs.

An important point to note is that this feature does not require the end systems to be Token Ring-attached. The remote end systems can connect over SDLC, Ethernet, or QLLC, and this feature will still work. The central site channel gateway must be LAN-attached (preferably Token Ring-attached). On Ethernet, however, duplicate MAC addresses for channel gateways will only work if you have a unique bridged Ethernet segment that are not bridged together, either in the router (by putting them in the same bridge group) or by a separate bridge. (Token Ring networks can rely on SRB for loop prevention.) You can locally load balance if the Ethernet segments are not bridged together in the router (by putting them in the same bridge group) or by a separate bridge group. This configuration is especially beneficial when you want to use duplicate network interface card (NIC) addresses with Ethernet. In 3-2, Router A has two reachability cache entries for MAC address 4000.000.0001 pointing to different bridge groups. Router A will load balance link-establish requests in round-robin mode because it is configured with the `dlsw load-balance round-robin` command.

Figure 3-2 DLSw+ Doing Local Load Balancing with Duplicate NIC Addresses in an Ethernet Environment

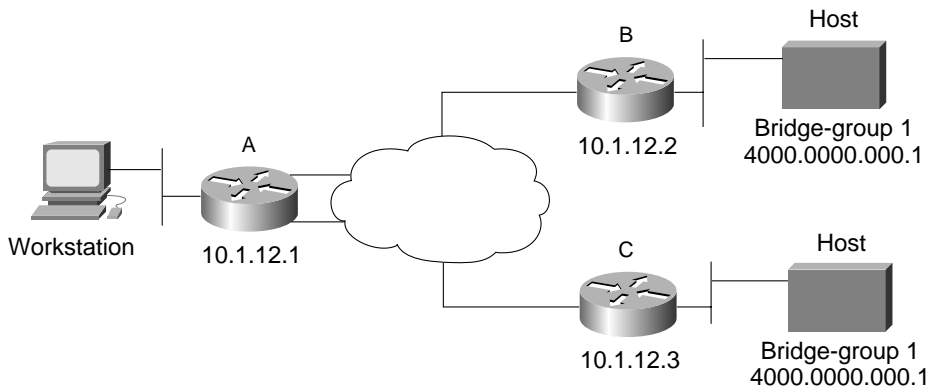


```
Configuration for Router A
dsw local-peer peer id 10.1.12.2
dsw bridge-group 1
dsw bridge-group 2
bridge 1 protocol ieee
dsw load-balance round-robin
int e0
  no ip address
  bridge-group 1
bridge 2 protocol ieee
int e1
  no ip address
  bridge-group 2
```

Alternately, if you are running Cisco IOS Release 12.1 or later, you can enable the Ethernet Redundancy feature. See the “Ethernet Redundancy” chapter.

You can do remote load balancing with duplicate NIC addresses in an Ethernet environment if the host devices with the duplicate NIC address are not sharing the same Ethernet LAN segment. In 3-3, Router A has two reachability cache entries for MAC address 4000.000.0001 pointing to different peer addresses. Note that the bridge groups are the same.

Figure 3-3 DLSw+ Doing Remote Load Balancing with Duplicate NIC Addresses in an Ethernet Environment



```
Configuration for Router A
dlsw local-peer peer id 10.2.12.1
dlsw remote-peer 0 tcp 10.1.12.2 circuit weight 10
dlsw remote-peer 0 tcp 10.1.12.3 circuit weight 10
dlsw remote-peer 0 tcp 10.2.20.1
dlsw load-balance circuit-count
dlsw timer explorer-wait-time 100
```

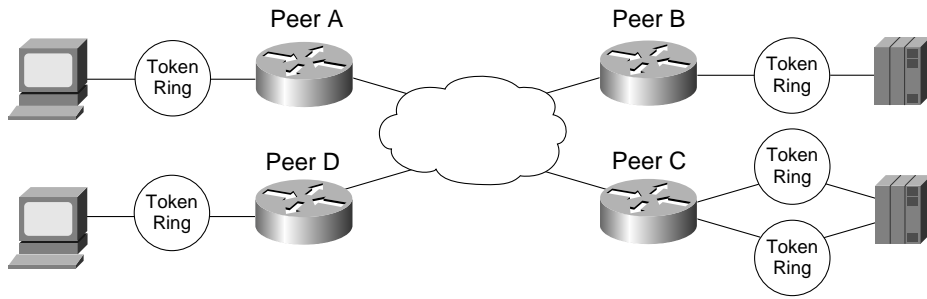
```
Configuration for Router B
dlsw local-peer peer id 10.1.12.2 cost 2
dlsw bridge-group 1
bridge 1 protocol ieee
interface e1
no ip address
bridge-group 1
```

```
Configuration for Router C
dlsw local-peer peer id 10.1.12.3 cost 2
dlsw bridge-group 1
bridge 1 protocol ieee
interface e1
no ip address
bridge-group 1
```

Router A will load balance equally between Peer B and C because the `dlsw load-balance circuit-count` command is configured and because equal values are specified in the `circuit-weight` of the `dlsw remote-peer` commands. See the “Load Balancing” section of this chapter for more details on how to configure load balancing.

An alternate way to specify cost is to use the `dlsw remote-peer` command as shown in Figure 3-4. Specifying cost in the `dlsw remote-peer` commands allows different divisions or parts of the country to favor different central site gateways. In addition, you must specify cost if you want to split SNA traffic across multiple central site routers, but each remote site has only a single SNA PU (all logical unit [LU] sessions flow over the same circuit that the PU session flows over). In Figure 3-4, Peer A always favors Peer B and Peer D always favors Peer C.

Figure 3-4 Configuration Where Cost Is Specified in the `dlsw remote-peer` Command instead of the `dlsw local-peer` Command



```

Configuration for Peer A
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2 cost 2
dlsw remote-peer 0 tcp 10.2.24.3 cost 4
    
```

```

Configuration for Peer B
dlsw local-peer peer-id 10.2.24.2
promiscuous
    
```

```

Configuration for Peer D
dlsw local-peer peer-id 10.2.18.6
dlsw remote-peer 0 tcp 10.2.24.2 cost 4
dlsw remote-peer 0 tcp 10.2.24.3 cost 2
    
```

```

Configuration for Peer C
dlsw local-peer peer-id 10.2.24.3
promiscuous
dlsw duplicate-path-bias load-balance
    
```

Load Balancing Mode

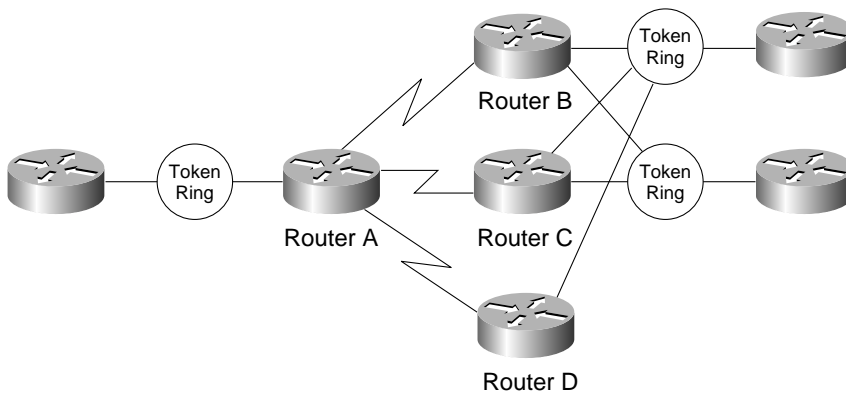
If the `dlsw load-balance` command is configured, DLSw+ load balances multiple paths based on whether the `round-robin` or `circuit-count` keyword is selected. If `round-robin` is specified, a peer distributes new circuits in a round-robin fashion to the capable peers (peers that have the lowest or equal cost specified) in the cache. DLSw+ load balances between two paths with the lowest cost if they are the lowest known. For example, in Figure 3-5, suppose the user wants to load balance between Routers B and C and that each are configured with a cost of 3. If Router D has a cost of 2, however, all circuits will establish through Router D because it is the lowest known cost.

In Figure 3-5, a workstation on the left sends an explorer packet to Peer A. Peer A forwards the explorer to both Peer B and Peer C. Peer B and Peer C forward the explorer on their local LAN. Peer B and Peer C receive a positive reply to the explorer and send a positive response back to Peer A. Peer A records that it has two peers it can use to reach the MAC address of the SNA device.

Peer A forwards a positive response to the workstation and then establishes an end-to-end circuit using Peer C or Peer B (depending on the first response received). Peer A distributes any new circuits between Peer B and Peer C in a round-robin fashion.

If, for example, Router B fails, all SNA sessions are terminated and reestablished through Router C. When Router B becomes available again, the sessions remain active on Router C (despite Router B's recovery.) Router A distributes any new circuits in a round-robin mode between Router B and Router C.

Figure 3-5 DLSw+ with Enhanced Load Balancing/Round-Robin Mode



```

Router A
dlsw local-peer peer-id 10.2.19.1
dlsw remote-peer 0 tcp 10.2.24.2
dlsw remote-peer 0 tcp 10.2.19.5
dlsw remote-peer 0 tcp 10.2.20.1
dlsw load-balance round-robin
dlsw timer explorer-wait-time 100

Router B
dlsw local-peer peer-id 10.2.24.2 cost 1 promiscuous

Router C
dlsw local-peer peer-id 10.2.19.5 cost 1 promiscuous

Router D
dlsw local-peer peer-id 10.2.20.1 promiscuous
    
```

Note: In Cisco IOS Release 12.0(3)T, the `dlsw load-balance` command replaced the `dlsw duplicate-path-bias load-balance` command. Although the `dlsw duplicate-path-bias load-balance` command continues to be accepted, it is converted to the new command if the configuration is displayed or saved. As with the `dlsw duplicate-path-bias load-balance` command, how the cache entries are used depends on which keyword (`round-robin` or `circuit count`) is specified.

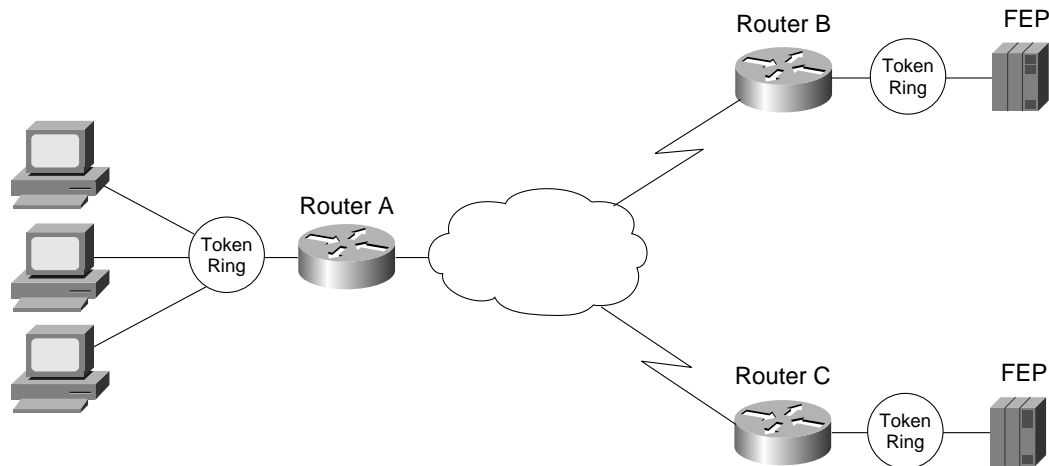
If `circuit count` is specified, then the Enhanced Load Balancing feature is configured. Each new circuit gets distributed based on existing loads and the desired ratio. The user assigns a `circuit-weight` value to the local peer and to each remote capable peer to create a desired ratio among the peers. The DLSw+ Enhanced Load Balancing feature calculates the difference between the desired and the actual ratio of circuits being used on a peer. It detects the path that is underloaded in comparison to the other capable peers and assigns new circuits to that path until the desired ratio is achieved.

The Enhanced Load Balancing feature load balances among peers and local ports; however, the `circuit weight` can only be applied to peers. DLSw+ distributes the traffic among local ports in a round-robin fashion, even if the user configures them for `circuit weight`.

In Figure 3-6, Router A, B, and C have a *circuit weight* of 10. In this case, there is 1:1 ratio between Router B and Router C and therefore, Router A knows that Router B and Router C should be handling the same number of circuits. Router A distributes the circuits evenly between Router B and Router C.

If, for example, Router B fails, all SNA sessions are terminated and reestablished through Router C. When Router B becomes available again, the sessions remain active on Router C (despite Router B's recovery.) Router A distributes any new circuits to router B until it has the same number of circuits as Router C, achieving the 1:1 ratio with Router C. Compare this to the round-robin method where Router A would alternate the new circuits between Router B and Router C, resulting in an imbalance in circuit distribution.

Figure 3-6 DLSw+ with Enhanced Load Balancing/Circuit Count Mode



```

Router A
dlsw local-peer peer-id 10.2.19.1
dlsw remote-peer 0 tcp 10.2.24.2 circuit weight 10
dlsw remote-peer 0 tcp 10.2.20.1 circuit weight 10
dlsw load-balance circuit-count
dlsw timer explorer-wait-time 100

Router B
dlsw local-peer peer-id 10.2.24.2 cost 1 promiscuous

Router C
dlsw local-peer peer-id 10.2.19.5 cost 1 promiscuous

```

Suppose the configuration was the following:

```

Router A
dlsw local-peer peer-id 10.2.19.1
dlsw remote-peer 0 tcp 10.2.24.2 circuit weight 20
dlsw remote-peer 0 tcp 10.2.20.1 circuit weight 10
dlsw load-balance circuit-count
dlsw timer explorer-wait-time 100

Router B
dlsw local-peer peer-id 10.2.24.2 cost 1 promiscuous

Router C
dlsw local-peer peer-id 10.2.19.5 cost 1 promiscuous

```

In this case, there is 2:1 ratio between Router B and Router C because they are configured with a circuit-weight of 20 and 10, respectively. Router A learns that Router B should be handling twice as many circuits as Router C. Router A checks how many circuits it has with each peer and makes its decision based on a 2:1 ratio.



If, for example, Router B fails, all SNA sessions are terminated and reestablished through Router C. When Router B becomes available again, the sessions remain active on Router C (despite Router B's recovery.) Router A distributes any new circuits to Router B until it achieves the 2:1 ratio between Router B and Router C.

Controlling Peer Selection

A higher-cost peer can be used for a connection even when the lower-cost peer is active, if the higher-cost peer responds to the explorer before the lower-cost peer. If your network configuration allows this possibility, you can prevent it by adjusting a timer.

Setting the `dlsw timers explorer-wait-time` command causes DLSw+ to wait the specified amount of time before selecting a peer to use for connections. See how to modify timers in the next chapter. This timer can be set in Cisco IOS Release 11.0 and later. Prior to Release 11.0, this timer did not exist.

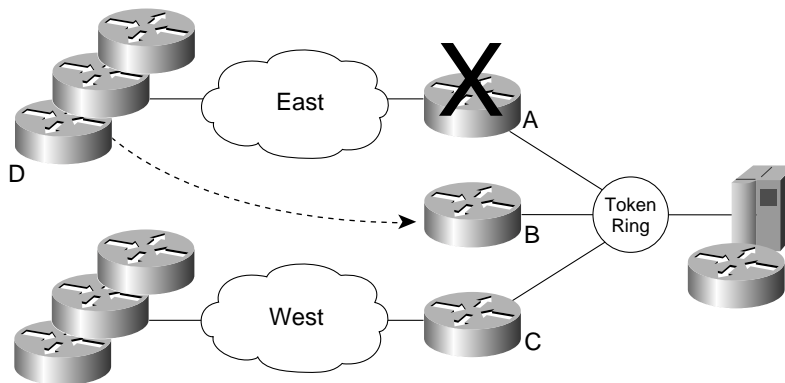
Backup Peers

Failure

Having multiple active peers is one way to provide dynamic and immediate recovery from the loss of a central site router. However, in some configurations you may prefer the alternate peer to be active only when required. This may be the case when the backup router resides at a disaster recovery site, or when there are more than 300 to 400 remote sites and a single central site router is providing backup for multiple central site routers.

In this case, use the backup peer capability (first available in Cisco IOS Release 10.3, but enhanced in Release 11.1). Figure 3-7 illustrates how to configure a backup peer. To use backup peers, any encapsulation method used to access the primary peer will work.

Figure 3-7 How to Use Backup Peers to Enhance Availability in a Large DLSw+ Network



```
Configuration for Router D
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2 -----> Router A is the primary
dlsw remote-peer 0 tcp 10.2.24.3 backup-peer 10.2.24.2 linger 20----> Router B to backup Router A
```

```
Configuration for Router A
dlsw local-peer peer-id 10.2.24.2 promiscuous
```

```
Configuration for Router B
dlsw local-peer peer-id 10.2.24.3 promiscuous
```

```
Configuration for Router E
dlsw local-peer peer-id 10.2.18.1 promiscuous
dlsw remote-peer 0 tcp 10.2.24.5 backup 10.2.24.2----> Router B to backup Router C
```

In this example, there are 400 remote sites. All the routers on the East Coast use Router A as the primary router, and all the routers on the West Coast use Router C as the primary router. In either case, the backup router is Router B. The configuration shown is the configuration in Router D, an East Coast router. (All the East Coast routers have the same two dlsw remote-peer commands.) Both the primary router (Router A) and the backup router (Router B) are configured in dlsw remote-peer commands. Router B is configured as a backup only, and the IP address of the router it is backing up is specified.

In the event of a failure in Router A, all SNA (or NetBIOS) sessions are terminated and reestablish through Router B. When Router A becomes available again, all new sessions are established through Router A, but sessions active on Router B remain on Router B until the linger timer expires. No new sessions (no explorers passed or new circuits established) are brought up on Router B during the configured linger period (20 minutes in this example). When the linger period expires, the backup peer connection is taken down.

Configuring a linger interval can be beneficial in some enterprises to allow network operators enough time to notify end users that active sessions established over the backup peer will be disrupted. It can also be used to minimize line costs if the backup peer is established over a dial connection.

A linger value of 0 must be configured on the backup remote-peer command to force the backup peer to drop immediately after the primary peer is reestablished. If no linger value is coded, the default behavior is to leave the backup peer up as long as active circuits remain connected over it, but not to allow new connections or active circuits over the backup peer when the primary peer recovers.

Note: Prior to Cisco IOS Release 11.1, when the primary peer was activated again, all sessions using the backup peer were terminated immediately and reestablished over the primary router. If that is not the action you want to take, and you are running a level of Cisco IOS Software earlier than Release 11.1, consider using duplicate active peers instead (described in the previous section).

Backup Peers Compared to Multiple Active Peers

Backup peers and multiple active peers (with one preferred and others capable) are two ways to ensure that a capable peer can back up the failure of a primary peer. One of the key differences in backup peers is that the peer connections are not active until they are needed. Suppose you have 1000 branch offices, and you want to design a network at minimal cost that will recover dynamically from the failure of any single central site router. Assume four routers at the central site can handle your traffic load. You can install four primary routers at the central site and define 250 branches to peer to each central site router.

To address your availability requirement, one option is multiple concurrently active peer connections. In this case, you would configure each remote router to have two peer connections, one to a preferred router and one to a capable router. The preferred router is the router configured with lower cost. The capable router can be the same router for all remote sites, but in that case, it would have 1000 peer connections. The largest number of peering routers we have seen is 400, and that was in an environment with extremely low traffic. Although 1000 idle peer connections are conceivable, as soon as the capable router takes over for another router, those peer connections could put a strain on the router. The other alternative is to have multiple central site routers as capable routers, but this is not the most cost-effective design.

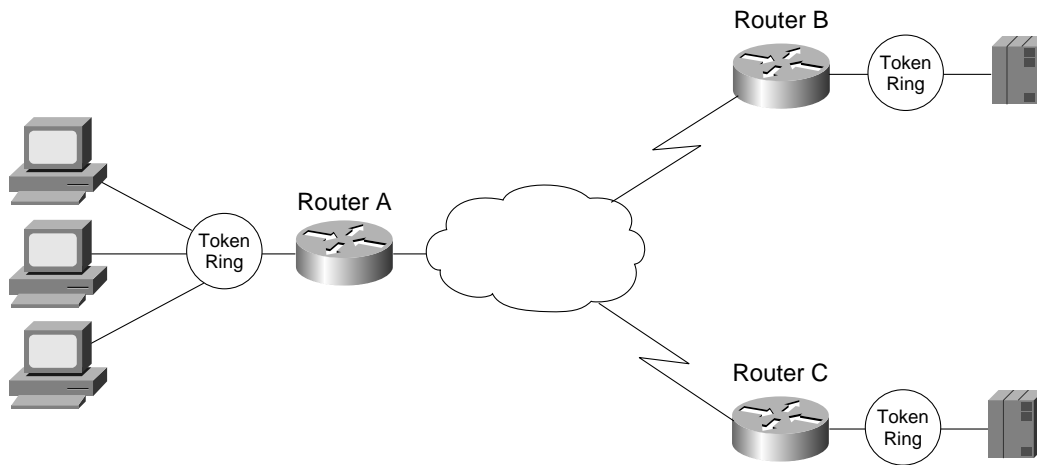
By using a backup peer statement in each remote branch instead of concurrently peering to two routers, a single backup router at a central site can easily back up any other central site router. There is no work on a backup router until a primary router fails.

Backup peers can be used to recover *only* from the loss of a router. They cannot be used to recover from the loss of a mainframe or mainframe channel gateway. The reason is because they are only activated when the primary *peer* fails. To enable automatic recovery from the loss of a mainframe or channel gateway, you must configure multiple active peers.

Summary of Availability Options

DLSw+ provides several options to enhance availability: load balancing, redundancy, and backup peers. Each option affects the distribution of circuits depending on the phase of operation (start-up, normal, failure, recovery from failure). Table 3-1 summarizes each availability option in DLSw+ and its effect on circuit distribution during the different phases of operation. The coordinate values (x/y) represent the number of circuits on Router B and Router C (Router B/Router C) in Figure 3-8.

Figure 3-8 Sample DLSw+ Configuration for Table 3-1



```
Router A
dlsw local-peer peer-id 10.2.19.1
dlsw remote-peer 0 tcp 10.2.24.2 circuit weight 10
dlsw remote-peer 0 tcp 10.2.20.1 circuit weight 10
dlsw load-balance circuit-count
dlsw timer explorerer-wait-time 100
```

```
Router B
dlsw local-peer peer-id 10.2.24.2 cost 1 promiscuous
```

```
Router C
dlsw local-peer peer-id 10.2.19.5 cost 1 promiscuous
```

Table 3-1 Sample Circuit Distribution in DLSw+

	Start up (Initial 8 Circuits)	Failure (Number of Circuits When Router A Fails)	Failure (Distribution of Recovered Circuits)	Recovery from Failure (Number of Circuits After Router A Recovers)	Normal (8 New Circuits)	Normal (8 New Circuits)
Backup Scenario	8/0	0/0	0/8	8/0 ¹	16/0	24/0
Fault-Tolerant Mode	8/0 ²	0/0	0/8	0/8	8/8	16/8
Load Balancing Round-Robin	4/4	0/4	0/8	0/8	4/12	8/16

Table 3-1 Sample Circuit Distribution in DLSw+

	Start up (Initial 8 Circuits)	Failure (Number of Circuits When Router A Fails)	Failure (Distribution of Recovered Circuits)	Recovery from Failure (Number of Circuits After Router A Recovers)	Normal (8 New Circuits)	Normal (8 New Circuits)
Enhanced Load Balancing Circuit Count	4/4	0/4	0/8	0/8	8/8	12/12
Cost Configured	8/0	0/0	0/8	0/8	8/8	16/8

1. If linger option is set to 0, the 8 circuits will stay with Router B.
2. Assuming Router A is the first path in the cache

Encapsulation Options

DLSw+ offers four different encapsulation options. These options vary in terms of the processing path they use, their WAN overhead, and the media they support. The encapsulation options are TCP, TCP/IP with RIF Passthru, FST, direct, and LLC2.

TCP Encapsulation

TCP is the standard DLSw encapsulation method and is the only encapsulation method supported by RFC 1795. TCP offers the most functionality of the encapsulation options. It provides reliable delivery of frames and local acknowledgment. It offers nondisruptive rerouting around link failures. With TCP encapsulation, you can take advantage of DDR to dynamically dial additional bandwidth if the primary link reaches a preconfigured amount of congestion. In most environments, it is the recommended encapsulation because its performance is generally more than adequate, it offers the highest availability, and the overhead generally has no negative impact on response time or throughput.

TCP is process switched, so it uses more cycles than FST or direct encapsulation. A Cisco 4700 Series router running DLSw+ with TCP encapsulation can switch up to 8 Mbps of data, so TCP encapsulation addresses the processing requirements of most SNA environments. Where higher throughput is required, additional routers or alternate encapsulation options can be used.

TCP encapsulation adds the most overhead to each frame (20 bytes for TCP and 20 bytes for IP in addition to the 16-byte DLSw header). TCP header compression or payload compression can be used to reduce the amount of bandwidth required, if necessary. At 56-kbps or higher line speeds, the 40 bytes of overhead adds less than 11 milliseconds to the round trip delay, so its impact is negligible.

DLSw+ with TCP encapsulation provides local acknowledgment and polling and minimizes keepalive traffic across the WAN. It supports any local and WAN media. See Appendix B, "DLSw+ Support Matrix," for supported media types. Load balancing across multiple WAN links or IP paths is possible because TCP resequences traffic.

When using TCP encapsulation, you can assign different types of traffic to different TCP ports so that queuing can be granular. LLC2 traffic can be distinguished by SAP (to distinguish NetBIOS and SNA traffic) and SNA devices can be prioritized by LOCADDR or a MAC/SAP pair.

The following is a sample dlsw remote-peer command specifying TCP encapsulation:

```
dlsw remote-peer 0 tcp 10.2.24.3
```

TCP/IP with RIF Passthru Encapsulation

TCP/IP with RIF Passthru is an option used to support multiple active paths between FEPs. It disables local acknowledgment and provides the packet with a complete end-to-end RIF, which replaces any state-based information that is normally required by DLSw+ to route packets. It is process switched and offers nondisruptive rerouting around link failures.

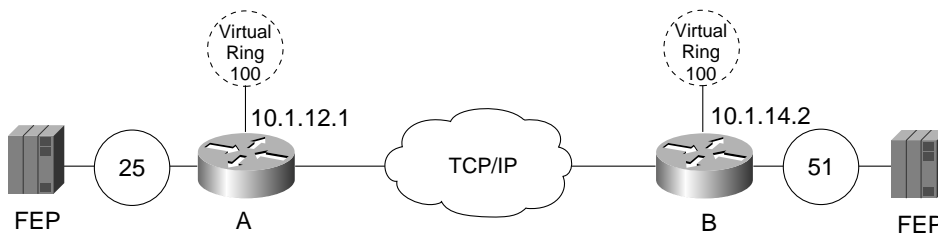
It is supported only when the end systems reside on Token Ring and are configured for SRB. See Appendix B, “DLSw+ Support Matrix,” for details.

TCP/IP with RIF Passthru supports activating a FEP via an NCP if the image is already loaded on its internal disk; otherwise, remote load is not supported. The following features are not supported with the DLSw+ RIF Passthru feature:

- Border peers
- Peer-on-demand peers
- Dynamic peers
- Backup peers

Prior to this feature, the design in Figure 3-9 was not supported:

Figure 3-9 Unsupported DLSw+ Configuration Prior to TCP/IP with RIF Passthru



The following is a sample `dlsw remote-peer` command specifying TCP/IP with RIF Passthru encapsulation:

```
dlsw remote-peer 0 tcp 10.2.24.5 rif-passthru 100
```

The virtual ring numbers must match between the DLSw+ peers. The Token Ring numbers, however, should be uniquely different throughout the network

FST Encapsulation

FST is a high-performance option used over higher-speed links (256-kbps or higher) when high throughput is required. FST uses an IP header with sequencing numbers to ensure that all frames are delivered in sequence (out-of-order frames are discarded and the end system must retransmit them).

FST is fast switched, not process switched, so using this encapsulation allows DLSw+ to process more packets per second than TCP encapsulation. FST does not use TCP, so the header is 20 bytes smaller.

FST, however, provides neither reliable delivery of frames nor local acknowledgment. All keepalive frames flow end to end. FST does not support all media types. See Appendix B “DLSw+ Support Matrix” for details. FST will reroute around link failures, but rerouting may be disruptive because of LLC2 time-outs. In addition, load balancing across multiple WAN links or IP paths is not recommended with FST because frames may arrive out of order and FST will discard them, causing end systems to retransmit and reducing overall network performance.

Finally, queuing is not as granular with FST because you cannot assign different types of traffic to different TCP ports. This means that when using FST encapsulation, queuing algorithms cannot be distinguished by SAP (so NetBIOS and SNA are treated as LLC2 traffic), and they cannot be distinguished by LOCADDR or MAC address.

The following is a sample `dls w remote-peer fst` command specifying FST encapsulation:

```
dls w remote-peer 0 fst 10.2.24.3
```

Direct Encapsulation

Direct encapsulation is a minimal-overhead option for transport across point-to-point lines where rerouting is not required. Direct encapsulation is supported over HDLC lines and Frame Relay. It includes a DLSw 16-byte header and the data-link control header.

Direct encapsulation is fast switched, not process switched, so using this encapsulation allows DLSw+ to process more packets per second than TCP encapsulation.

Direct encapsulation provides neither reliable delivery of frames nor local acknowledgment. All keepalive frames flow end to end. Direct encapsulation does not support all media types. See Appendix B, “DLSw+ Support Matrix,” for details. Direct encapsulation does not provide any rerouting.

Finally, queuing is not as granular with direct encapsulation because you cannot assign different types of traffic to different TCP ports. This means that when using direct encapsulation, queuing algorithms cannot be distinguished by SAP (so NetBIOS and SNA are treated as LLC2 traffic), and they cannot be distinguished by SDLC or MAC address.

Direct encapsulation is sometimes considered for very low-speed lines to minimize overhead, but TCP encapsulation with payload compression may offer lower WAN overhead without the limitations of direct encapsulation.

The following is a sample `dls w remote-peer interface` command specifying direct encapsulation on an HDLC line:

```
dls w remote-peer 0 interface serial 01
```

The following is a sample `dls w remote-peer frame relay` command specifying direct encapsulation on a Frame Relay line:

```
dls w remote-peer 0 frame-relay interface serial 01 33 pass-thru  
int s1  
frame-relay map dls w 33
```

In this example, data-link connection identifier (DLCI) 33 on serial interface 1 will be used to transport DLSw+ traffic. Specifying `pass-thru` implies that the traffic is not locally acknowledged. Leaving `pass-thru` off will cause the traffic to be locally acknowledged, which means it is transported in LLC2 to ensure reliable delivery. The next section describes LLC2 encapsulation.

LLC2 Encapsulation (DLSw Lite)

DLSw+ with LLC2 encapsulation is also known as DLSw Lite. It supports many DLSw+ features, including local acknowledgment, media conversion, minimizing keepalive traffic, and reliable delivery of frames, but it uses less overhead (16 bytes of DLSw header and 4 bytes of LLC2). It is currently supported only over Frame Relay and assumes a point-to-point configuration over Frame Relay (that is, the peering router at the central site is also the WAN router). DLSw Lite does not support all media types. See Appendix B, “DLSw+ Support Matrix,” for details. DLSw Lite is process switched and processes approximately the same traffic volume as TCP encapsulation.

With DLSw Lite, link failures are disruptive. Availability can be achieved by having multiple active central site peers, which allows for dynamic, but disruptive, recovery from the loss of either a link or a central site peer. Backup peers are supported for DLSw Lite in Cisco IOS Release 11.3.

Queuing with DLSw Lite is not as granular as with TCP encapsulation, because you cannot assign different types of traffic to different TCP ports. This means that when using DLSw Lite, queuing algorithms cannot distinguish traffic by SAP (so NetBIOS and SNA are treated as LLC2 traffic), and they cannot distinguish traffic by SDLC or MAC address.

The following is a sample dlsw remote-peer frame-relay command specifying LLC2 encapsulation on a Frame Relay line:

```
dlsw remote-peer 0 frame-relay interface serial 01 33
int s1
  frame-relay map llc2 33
```

Note: The frame-relay map llc2 command will not work on point-to-point sub-interfaces. Instead, you must provide the DLCI number in the frame-relay interface-dlci command and specify the same DLCI number in the dlsw remote-peer frame relay command as follows:

```
dlsw remote-peer 0 frame-relay interface serial 0 60
  interface s0.1 point-to-point
  frame-relay interface-dlci 60
```

Encapsulation Overhead

Different types of encapsulation incur different amounts of overhead on a per-frame basis. But with TCP and LLC2, local acknowledgment and keepalive traffic are removed from the WAN, reducing the number of packets. Also, techniques like payload or header compression and packing multiple SNA frames in a single TCP packet can further reduce the overhead. The percentage of overhead created by DLSw depends on the encapsulation method used.

Figure 3-10 illustrates the frame format for TCP, FST, DLSw Lite, and direct encapsulation. The percentage shown is the amount of overhead assuming SNA transactions of 40 in, 1920 out (a screen refresh) and 40 in, 1200 out. With smaller transactions the overhead is larger. The TCP encapsulation numbers are worst-case numbers because they assume that each SNA path information unit (PIU) is encapsulated in a separate TCP packet. In fact, if there is more than one SNA PIU in the output queue, multiple frames will be encapsulated in a single TCP packet, reducing the overhead. The percentages in Figure 3-10 do not take into consideration the fact that DLSw+ eliminates keepalive packets and acknowledgments.

Figure 3-10 Frame Format and Per-Packet Overhead of Various Encapsulation Types and Transaction Sizes

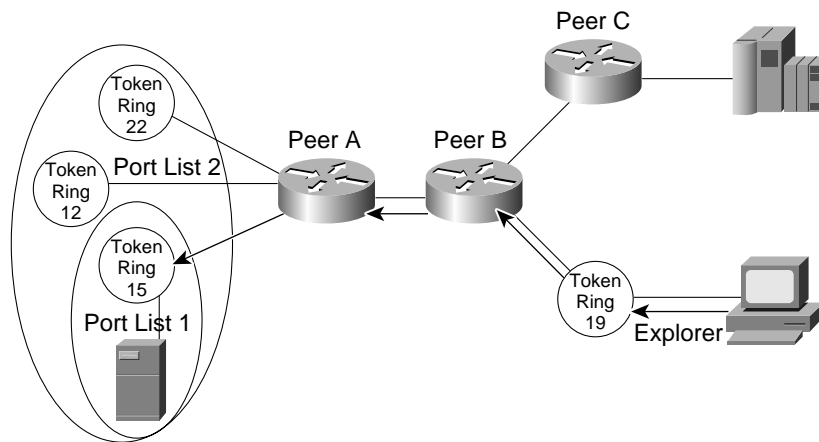
Encapsulation	40/1920		40/1200	
	SDLC	LAN	SDLC	LAN
TCP	5.7%	4.5%	9%	7%
FST	3.7%	2.4%	5.8%	3.9%
DLSw Lite	2%	1%	3.2%	1.3%
Direct	1.8%	.6%	2.9%	1%

The effective per-packet overhead of DLSw for LAN traffic is lower than SDLC because DLSw+ eliminates the need to carry MAC addresses and RIFs in every frame. DLSw+ does not carry this data because the DLSw+ circuit ID (part of the 16-byte DLSw header) is used for circuit correlation. The overhead of MAC addresses and RIFs can range from 12 to 28 bytes of data. The percentages in Figure 3-10 assume the minimum overhead (no RIF).

Port Lists

Port lists allow you to create broadcast domains in a DLSw+ network. Using port lists, you can control where broadcasts are forwarded. For example, in Figure 3-11 there are three rings at the distribution site (where Peer A resides). All the rings have SNA end systems, but Ring 15 is the only ring with NetBIOS servers. The branch with Peer B needs access to the NetBIOS servers on Ring 15, but does not need access to other rings. Port lists allow you keep all broadcasts from Peer B off Rings 12 and 22 (and prevent Peer B from communicating with devices on Rings 12 or 22). You can distinguish among different Token Ring ports and serial ports using port lists, but all Ethernet ports are treated as a single entity (Ethernet bridge group).

Figure 3-11 Ring Lists Used to Limit Broadcast Domains in a DLSw+ Network



```

Configuration for Peer A
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 1 tcp 10.2.24.2 /* Peer B is associated with port list 1
dlsw remote-peer 2 tcp 10.2.24.3 /* Peer C is associated with port list 2
dlsw ring-list 1 rings 15
dlsw ring-list 2 rings 22 12 15

```

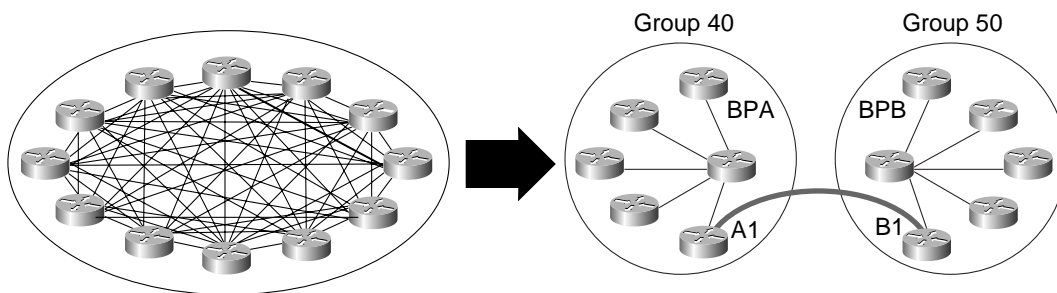
Peer Groups, Border Peers, Peer Group Clusters, and On-Demand Peers

Peer groups and border peers can be used to minimize the number of peer connections required for any-to-any communication. Prior to the introduction of border peers, any two DLSw+ routers that required connectivity needed a peer connection active at all times. This peer connection is used to find resources and to carry circuit traffic. In a fully meshed network of n routers, this requires $n \times (n-1)/2$ TCP connections. This is complex to configure and can result in unnecessary explorer traffic. To address this issue, DLSw+ supports the concept of peer groups and border peers. Peer groups are arbitrary groups of routers with one or more designated border peers. Border peers form peer connections with every router in their group and with border peers in other groups. The role of a border peer is to forward explorers on behalf of other routers.

The border peer's functionality was enhanced in Cisco IOS Release 11.3 with the Border Peer Caching feature. The border peers check their local, remote and group cache before forwarding explorers to other routers. The local cache gives the border peer reachability information on its local data-link control. If the border peer finds that it can reach a destination via its local cache, then it does not forward the explorer to other peers. The remote cache gives the border peer reachability information within its own peer group. If the border peer finds it can reach a destination via its remote cache, then it forwards the explorer only to that peer. The group cache gives a border peer reachability information about other peer groups to which it does not belong. If the border peer finds it can reach a destination via its group cache, then it sends the explorer only to a border peer in that specific group.

In Figure 3-12, the “before” network shows the required TCP connections for fully meshed connectivity without using border peers. Without border peers, any time a router wants to find a resource that is not in its cache, it must create an explorer frame and replicate it for each TCP connection. This creates excessive explorer traffic on the WAN links and processing load on the router.

Figure 3-12 Using Border Peers and Peer Groups to Minimize the Number of Required TCP Connections while Maintaining Full Any-to-Any Connectivity



```
Configuration for Peer A1
dlsw local-peer peer-id 10.2.17.1 group 40 promiscuous
dlsw remote-peer 0 tcp 10.2.24.1
dlsw peer-on-demand-defaults tcp
```

```
Configuration for Border Peer A
dlsw local-peer peer-id 10.2.24.1 group 40
border promiscuous
dlsw remote-peer 0 tcp 10.2.18.2
```

```
Configuration for Peer B1
dlsw local-peer peer-id 10.2.24.3 group 50 promiscuous
dlsw remote-peer 0 tcp 10.2.18.2
dlsw peer-on-demand-defaults tcp
```

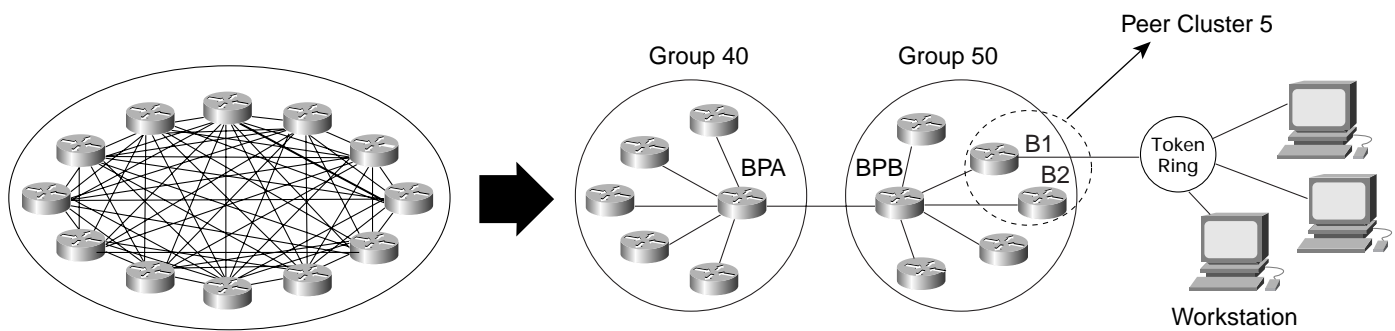
```
Configuration for Border Peer B
dlsw local-peer peer-id 10.2.18.2 group 50
border promiscuous
dlsw remote-peer 0 tcp 10.2.24.1
```

After configuring border peers and peer groups, the same fully meshed connectivity is possible without the overhead. In the “after” network, two peer groups are defined (Group 40 and Group 50). Within each group, one or more peers are configured as border peers. Every peer within Group 40 establishes a peer connection with border peer A (BPA). Every peer within Group 50 establishes a peer connection with border peer B (BPB). The border peers establish a peer connection with each other. When a peer in Group 40 wants to find a resource, it sends a single explorer to its border peer. The border peer checks its local, remote and group cache. If the resource is located in one of its caches, then the border peer forwards the explorer to the destination. If the resource is not located in the border peer's cache, the border peer forwards this explorer to every peer in its group and to every other border peer. BPB, after receiving this explorer, forwards it to every peer in its group. When the resource is found (in this case at B1), a positive reply flows back to the origin (A1) via the two border peers. At this point A1 establishes a direct peer connection to B1. Peer connections that are established via border peers without the

benefit of preconfiguration are called peer-on-demand connections. The rules for establishing on-demand peers are defined in the `dls w peer-on-demand-defaults tcp` command in each router. Use peer groups and border peers only when you need branch-to-branch communication between NetBIOS or APPN end systems.

The Peer Group Cluster feature was introduced in Cisco IOS Release 12.0(3)T to further minimize explorer replication in border peer networks. Peer group clusters can be used in very large border peer networks where multiple routers within a peer group are serving the same LAN. DLSw+ “clusters” DLSw+ peers that are connected to the same LAN into logical groups. When the multiple peers are defined in the same peer cluster, the DLSw+ border peer does not forward explorers to more than one member within the same peer cluster. In Figure 3-13, member peers B1 and B2 are serving the same Token Ring LAN and have been configured into Peer Cluster 5.

Figure 3-13 Using Border Peers, Peer Groups, and Peer Clusters to Minimize the Number of Required TCP Connections while Maintaining Full Any-to-Any Connectivity



Configuration for Peer B2

```
dls w local-peer peer-id 10.2.24.5 group 50 promiscuous cluster 5
dls w remote-peer 0 tcp 10.2.18.2
dls w peer-on-demand-defaults tcp
```

Configuration for Peer B1

```
dls w local-peer peer-id 10.2.24.3 group 50 promiscuous cluster 5
dls w remote-peer 0 tcp 10.2.18.2
dls w peer-on-demand-defaults tcp
```

Configuration for Border Peer A

```
dls w local-peer peer-id 10.2.24.1 group 40
  border promiscuous
dls w remote-peer 0 tcp 10.2.18.2
```

Configuration for Border Peer B

```
dls w local-peer peer-id 10.2.18.2 group 50
  border promiscuous
dls w remote-peer 0 tcp 10.2.24.1
```

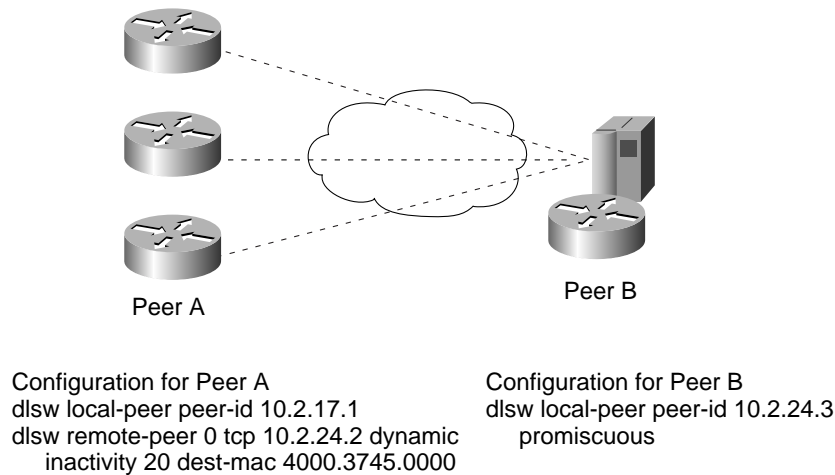
Dynamic Peers

Dynamic peers (available in Cisco IOS Release 11.1 and later) are configured remote peers that are connected only when there are circuits using them. When a `dls w remote-peer` command specifies `dynamic`, the remote peer is activated only when an end system sends an explorer frame that passes all the filter conditions specified in the `dls w remote-peer` command. When the dynamic peer connection is established, the explorer is forwarded to the remote peer. If the resource is found, a circuit is established and the remote peer remains active until all circuits using that remote peer terminate and ten minutes elapse. You can specify the `no-llc` keyword to modify the elapsed time to something other than ten minutes. Optionally, the remote peer can be configured to disconnect when there is no activity on any of the circuits for a prespecified amount of time (inactivity *minutes*).

Filters that minimize how many explorers are sent to a remote peer can be included in `dls w remote-peer` commands. In the case of dynamic peers, these filters are also used to prevent the dynamic peer from being activated. The `remote-peer` statement allows you to point to lists of SAPs, MAC addresses, NetBIOS names, or

byte offset filters. You can also specify a MAC address on the `dlsw remote-peer` command for a dynamic peer, in which case that remote peer is activated only when there is an explorer for the specified MAC address. Figure 3-14 shows an example of how to use this feature. In Figure 3-14, the dynamic peer is only established if an explorer frame is received that is destined for the MAC address of the FEP. After the peer connection is established, if there is no activity on this peer connection for 20 minutes, the peer connection and any circuits using the connection are terminated because inactivity 20 was specified.

Figure 3-14 DLSw+ Routers Configured to Take Advantage of the Dynamic Peer Feature



When to Use Dynamic Peers

Use dynamic peers if you have a large network but do not require all remote sites to be connected at the same time. By using dynamic peers, you can minimize the number of central site routers needed to support the network. You can also use dynamic peers for occasional communication between a pair of remote sites. Dynamic peers differ from on-demand peers because they must be preconfigured. Finally, for small networks, dynamic peers can be used to dial out during error recovery.

SNA Dial-on-Demand Routing

SNA DDR refers to the ability for DLSw+ to transfer SNA data over a dial-up connection and automatically drop the dial connection when there is no data to send. The SNA session remains active. To use SNA DDR, configure the following on the `dlsw remote-peer` command:

```

dlsw remote-peer list-number tcp ip-address dynamic keepalive 0 timeout seconds
  [inactivity seconds dmac-out mac-address tcp-timeout seconds]
  
```

The `dynamic` keyword is optional but recommended because it will prevent the remote peer connection from being established unnecessarily. The `dynamic` option is described in the previous section and can be used in conjunction with the `dmac-out` or `dmac-output-list` options on the `dlsw remote-peer` command to ensure that peer connections are only brought up when desired (for example, when a device is trying to locate the FEP).

The `keepalive` keyword is required. DLSw+ locally acknowledges SNA (or more precisely, SDLC or LLC2) traffic, so no data-link control acknowledgments or receiver ready frames bring up the dial connection. However, DLSw+ peers send peer keepalives to each other periodically, and these keepalives do bring up the dial connection. The `keepalive` option refers to how often DLSw+ peers send peer keepalives to each other. If you set this to zero, no keepalives are sent and, therefore, the peer keepalive does not keep the dial line up. You must

specify `keepalive 0` in *both* peers; that is, either you must specify the remote peers at both the local and remote DLSw+ routers, or you must use the `prom-peer-default` command to set `keepalive` to zero for all promiscuous peer connections. The `prom-peer-default` command has the same options as the `peer-on-demand-defaults tcp` command and is available in the later maintenance release of all DLSw+ releases.

The `keepalive` parameter refers to how often DLSw+ peers send peer keepalives to each other. If you set this to zero, no keepalives are sent, and hence the peer `keepalive` does not keep the dial line up. This parameter must be specified in *both* peers, which means that you must either specify the remote peers at both the local and remote DLSw+ routers, or you must use the `dlsw prom-peer-default` command to set `keepalive` to zero for all promiscuous peer connections. The `dlsw prom-peer-default` command is similar to the `dlsw peer-on-demand-defaults tcp` command and is available in the later maintenance releases of all DLSw+ releases.

The `timeout` keyword is recommended. Without peer keepalives, DLSw+ is dependent on TCP timers to determine when the SNA session has come down. TCP only determines that it has lost a partner if it does not get an acknowledgment after it sends data. By default, TCP may wait up to 15 minutes for an acknowledgment before tearing down the TCP connection. Hence, when `keepalive 0` is specified, you should also set the `timeout` keyword, which is the number of seconds that TCP waits for an acknowledgment before tearing down the connection. `Timeout` should be long enough to allow acknowledgments to get through in periods of moderate to heavy congestion, but short enough to minimize the time it takes to recover from a network outage. SNA data-link control connections typically wait 150 to 250 seconds before timing out.

Other Considerations

In addition to preventing keepalive traffic from bringing up the Integrated Services Digital Network (ISDN) lines, you need to worry about routing updates. In hub and spoke environments, to prevent route table updates from bringing up the dial connections, use static routes. Alternatively, you can use Routing Interface Protocol (RIP) Version 2 or on-demand routing (ODR) for IP routing from the dial-up branches to the central site. ODR is a mechanism that provides minimum-overhead IP routing for stub sites. Define RIP Version 2 or ODR on the ISDN interface of the central router as passive mode. Then redistribute RIP Version 2 or ODR routes into the main routing protocol (Enhanced Interior Gateway Routing Protocol [EIGRP] or Open Shortest Path First [OSPF]). This allows you to have multiple routers at the central site for load balancing or redundancy. Whichever router receives the call from the remote site has the route installed dynamically. At the remote site, the routing protocol (RIP or ODR) must be denied from the dialer list.

For meshed topologies, you can minimize routing table updates by using a distance-vector protocol such as RIP or IGRP in combination with Cisco's snapshot routing feature. Snapshot routing prevents regular routing updates from bringing up the ISDN connection. The changes in routing tables are sent either when the link is opened by end-user traffic or at a regular configurable interval. Snapshot routing supports not only IP routing updates, but also Novell's IPX routing and SAP updates.

Many NetBIOS implementations use a session keepalive (in addition to a data-link control keepalive) to maintain sessions, so DDR may not work with NetBIOS. (The session level keepalive will keep the dial line up.) To address this issue, a new capability was added in Cisco IOS Release 11.3. A new command, `dlsw netbios-keepalive-filter`, filters session keepalives and prevent them from bringing up the WAN link.

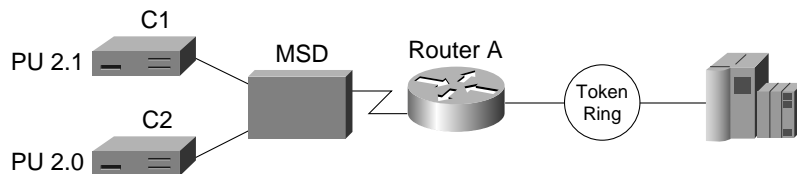
Local Switching

Local switching (available in Cisco IOS Release 11.1 and later) allows a single router to provide media conversion between SDLC and Token Ring and between QLLC and LAN. This is useful in environments that need simplified SNA network design and improved availability. For example, by converting SDLC to Token Ring, fewer FEP expansion frames are required; moves, adds, and changes are easier; and recovery from a FEP or Token Ring

interface coupler (TIC) failure can be automatic (by using duplicate TIC addresses). Local switching can be used to connect SDLC devices directly to a Cisco router with a CIP card. Local switching can also be used over a WAN where the remote branch has SNA devices on LANs, but the central site FEP still requires serial connectivity (for example, when the FEP is an IBM3725).

To use local switching, omit `dlsw remote-peer` commands. In the `dlsw local-peer` command, the peer ID is unnecessary. A sample network and configuration is shown in Figure 3-15.

Figure 3-15 Local Switching Configuration in a Mixed PU 2.0 and PU 2.1 Environment



```

Configuration for Router A
dlsw local-peer
source-bridge ring group 100
interface serial 0
...
sdlc role primary
sdlc vmac 4000.3174.0000
sdlc address c1 xid-poll
sdlc partner 4000.3745.0001 c1
sdlc address c2
sdlc xid c2 01767890
sdlc partner 4000.3745.0001 c2
sdlc dlsw c1 c2
interface tokenring 0
source bridge 1 1 100
source bridge spanning

```

