

Customization

This chapter describes several ways to customize your DLSw+ network. It includes a description of filtering and static device configuration options, as well as ways to tune network performance by controlling message sizes, timers, and queue depth. Each topic includes the router configuration changes, the effect of the changes, and the benefits that can be derived from the changes. These tuning and customization suggestions are not prerequisites for achieving good performance from DLSw+, but they offer a way to improve overall network performance. They are optional and are unnecessary in many environments.

Read this chapter if you have a very large network (thousands of SNA PUs), a high volume of NetBIOS broadcasts, or a high number of SNA transaction rates (greater than 200 transactions per second).

Note: Tuning modifications should only be made with Cisco's assistance (for example, system buffer tuning).

Filtering

Filtering can be used to enhance the scalability of a DLSw+ network. For example, filtering can be used to:

- Reduce traffic across a WAN link (especially important on very low-speed links and in environments with NetBIOS)
- Enhance the security of a network by controlling access to certain devices

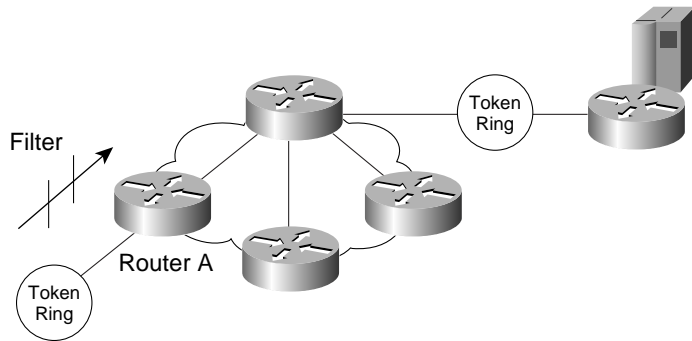
DLSw+ allows you to define access lists that are associated with a particular peer. This capability is powerful because it allows you to decide on a per-site basis what traffic should be allowed to pass over the network. These access lists use standard Cisco filter access list syntax.

To filter DLSw+ traffic on a remote peer basis, you must first define an access list containing the resources and the conditions for which you would like the router to pass traffic. You must then associate the access list to a remote peer.

The `dlsw remote-peer` command allows you to point to lists of SAPs, MAC addresses, NetBIOS names, or byte offset filters. You can also simply specify a MAC address in the `dlsw remote-peer` command. When these filters are specified, only explorers that pass the access list conditions are forwarded to the remote peer.

Figure 4-1 shows how to use filters to control traffic by protocol or SAP. In this example, the remote peer provides access to SNA resources but blocks all NetBIOS traffic from the WAN. NetBIOS workstations send out large numbers of broadcast frames that can easily overwhelm a low-speed WAN and cause throughput and connectivity problems. To prevent these problems, you can specify an access list as shown in Figure 4-1. The access list numbers can range from 200 to 299. Access lists are applied to peers in the `dlsw remote-peer` command.

Figure 4-1 Using Filtering to Control Traffic by SAP Type



```
Configuration for Router A
access-list 200 permit 0x0000 0x0d0d
dlsw remote-peer 0 tcp 10.17.24.12 lsap-output-list 200
```

Alternately, to allow NetBIOS and not SNA, specify:

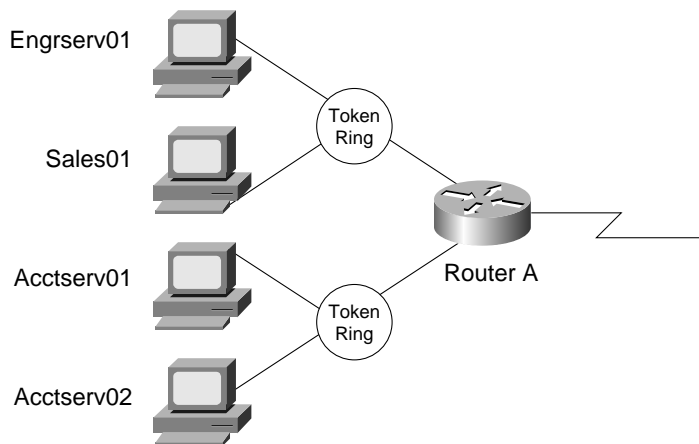
```
access-list 200 permit 0xf0f0 0x0101
```

Both access-list commands can be used to allow only SNA and NetBIOS traffic while blocking other SRB traffic, such as Novell IPX and TCP/IP, from being transmitted across the WAN by DLSw+.

Note: By default, DLSw+ handles all protocols not being routed by the router in which it resides.

Figure 4-2 shows the configuration required to allow any NetBIOS host with a name starting with “sales” to access the WAN, but not allow any other servers (for example, Engserv01 or Acctserv02) to access the WAN. This can be done for security reasons or to limit the traffic across the WAN link. By applying the access lists to the remote peers instead of the local interfaces, you allow traffic to be locally bridged.

Figure 4-2 Using Filtering to Limit the Broadcasts and Network Access of Individual NetBIOS Servers



```
Configuration for Router A
netbios access-list host salesfilt permit sales*
dlsw remote-peer 0 tcp 10.17.24.12 host-netbios-out salesfilt
dlsw peer-on-demand-defaults tcp host-netbios-out salesfilt
```

If you want to prevent this traffic from being forwarded by the router either locally or remotely, you can apply the filter to the Token Ring interface. To apply a NetBIOS access list to an interface, use the following command after the interface command:

```
netbios input-access-filter host name
```

Use this filter only if you need both local and remote filtering, because it will be applied to all locally bridged traffic and may impact local bridging performance.

Byte filters allow you to filter based on the content of arbitrary fields in a NetBIOS frame. The bytes list name (nblist) is the name of a previously defined NetBIOS bytes access list filter:

```
dlsw remote-peer 0 tcp 10.17.24.12 bytes-netbios-out nblist
```

Another technique to filter traffic is to specify the keyword `dest-mac` in the `dlsw remote-peer` command, which will allow only a single MAC address at the remote peer site to communicate to this local peer. Alternatively, the keyword `dmac-out` lets you specify an access list with multiple MAC addresses.

Static Configuration Options

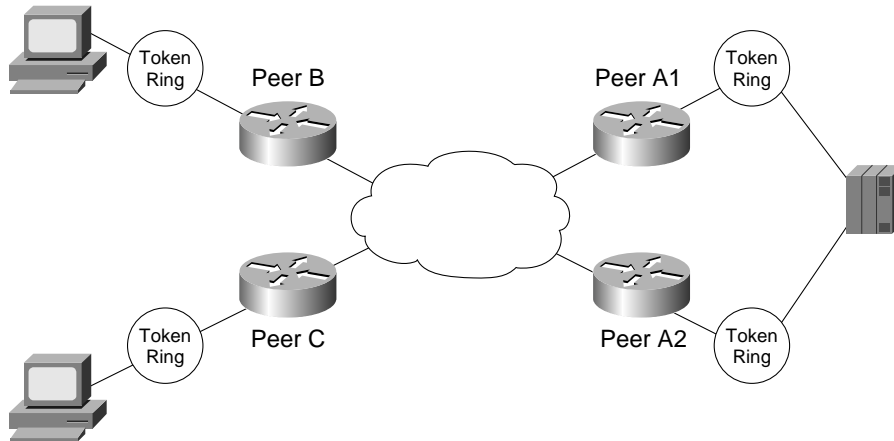
By predefining resources that are accessed frequently, you can minimize broadcast traffic. This traffic can be especially disruptive immediately following a failure of a key resource when every end system attempts to reconnect at the same time. DLSw+ allows you to predefine resources in two ways. You can configure local resources that you want a DLSw+ peer to advertise to other peers, or you can configure static paths that a peer will use to access remote resources.

Advertising Reachability

You can configure reachability of MAC addresses or NetBIOS names with a `dlsw icanreach` command. DLSw+ peers advertise this reachability to remote peers as part of the capabilities exchange. Figure 4-3 illustrates a way to use `dlsw icanreach` commands to prevent remote branches from sending any explorers destined for a mainframe channel gateway across the WAN. In Figure 4-3, two branch offices are shown with routers Peer B

and Peer C. At the data center, there are two central site routers, Peer A1 and Peer A2. Both data center routers advertise the reachability of the FEP to the remote routers as part of the capabilities exchange, allowing the branch routers to preload their cache with two paths to the MAC address of the FEP. After a major outage of a FEP or Token Ring, instead of having broadcasts flowing from each remote site, the remote sites will simply reconnect through the appropriate peer.

Figure 4-3 Hierarchical SNA Network Configured to Eliminate the Requirement for Explorers to Find the MAC Address of the FEP or a Mainframe Channel Gateway



```
Configuration for Peer B
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2
dlsw remote-peer 0 tcp 10.2.24.3
```

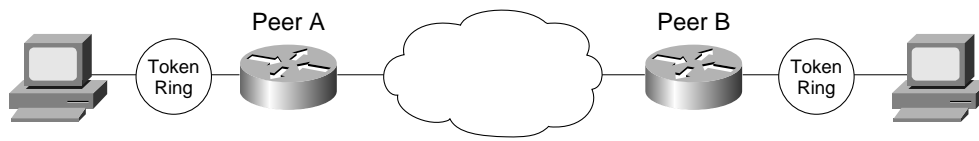
```
Configuration for Peer C
dlsw local-peer peer-id 10.2.18.1
dlsw remote-peer 0 tcp 10.2.24.2
dlsw remote-peer 0 tcp 10.2.24.3
```

```
Configuration for Peer A1
dlsw local-peer peer-id 10.2.24.2 cost 2
promiscuous
dlsw icanreach mac-addr 4000.3745.0001
```

```
Configuration for Peer A2
dlsw local-peer peer-id 10.2.24.3 cost 4
promiscuous
dlsw icanreach mac-addr 4000.3745.0001
```

The `dlsw icanreach` command also supports the `mac-exclusive` and `netbios-exclusive` keywords, which indicate that the resources advertised by this peer are the only resources the peer can reach. By specifying `mac-exclusive` or `netbios-exclusive`, you can indicate that the list of specified MAC addresses or NetBIOS names are the *only* ones reachable from a given router. Figure 4-4 shows how `dlsw icanreach netbios-exclusive` can be used to prevent other branch routers from sending explorers for NetBIOS servers other than those advertised.

Figure 4-4 DLSw+ Configured to Advertise Reachability of a Server While Concurrently Advertising that No Other NetBIOS Names Are Reachable



```

Configuration for Peer A
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.18.1
dlsw icanreach netbios-name nysales01
dlsw icanreach netbios-exclusive
    
```

```

Configuration for Peer B
dlsw local-peer peer-id 10.2.18.1
dlsw remote-peer 0 tcp 10.2.17.1
dlsw icanreach netbios-name lasales01
dlsw icanreach netbios-exclusive
    
```

Note that if you are using border peers, and remote branch routers do not establish peer connections between them, this reachability information is not exchanged (because the peer connection is not established until *after* the resource is found). When using border peers for branch-to-branch connectivity, sites that communicate frequently can configure direct peer connections and use the `dlsw icanreach` command to preload their cache entries. This eliminates the need to do broadcast searches for frequently accessed resources, but takes advantage of border peer dynamics to find infrequently accessed resources.

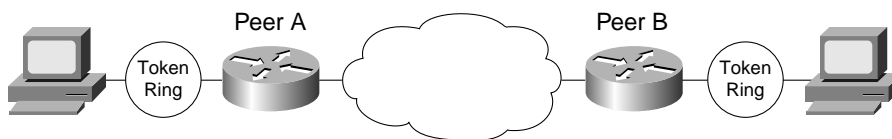
Reachability information learned as part of a capabilities exchange with a remote peer is considered valid as long as that remote peer is active. Multiple central site routers can advertise reachability of the same central site resources. If a remote branch router learns of multiple paths to a central site resource through the capabilities exchange, it will cache up to four paths, and the rules for duplicate path bias apply.

The `dlsw icanotreach saps` command allows you to list SAPs that this router cannot reach locally. This command can be used to advertise to a remote peer that it should not send explorers for certain SAPs (for example, NetBIOS). If there are only a few SAPs that this router can reach, it is probably easier to use the `dlsw icanreach saps` command.

Defining Static Paths

Static path definition allows a router to set up circuits without sending explorers (the entry is treated as stale until verified, however). The path specifies the peer to use to access a MAC address or NetBIOS name. The remote peer is identified by an IP address or an interface. Path information learned from a static path definition is never deleted. If a static path is not available, the circuit cannot be established. As a result, static paths are more appropriate if only one peer exists that can be used to access a remote node. Figure 4-5 shows how to configure a static path.

Figure 4-5 Configuration for a Static Path Always Used to Reach a Specified MAC Address



```

Configuration for Peer A
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.18.1
dlsw mac-addr 4000.0521.0001 ip-address 10.2.18.1
    
```

Table 4-1 compares the meaning and use of static path definitions to `icanreach` definitions.

Table 4-1 Comparison of Static Path Configuration and `icanreach` Configuration

Static Paths	icanreach
Defines paths to local or remote resources	Defines reachable local resources
Exclusive does not apply	Includes an exclusive option to minimize unnecessary broadcasts
Not exchanged with capabilities exchange used by this peer	Exchanged with capabilities exchange; used by remote peers
Never deleted	Deleted from cache when remote peer connection comes down
Multiple static paths possible	Multiple paths possible

Setting Transmission Unit Size

Controlling the size of the data transmitted across the network can affect performance in some situations. There are two features in the Cisco IOS Software that you need to consider: largest frame size and IP maximum transmission unit (MTU) path discovery.

Largest Frame Size

When a station is installed on a Token Ring, it can be configured to support a maximum frame size. When this device attempts to connect to its partner (for example, the server, CIP, or FEP), it must send an explorer to locate this device. The originator puts its maximum supported frame size in the explorer. The destination adjusts the maximum frame size before responding. When the response to the explorer is sent, each source-route bridge and each DLSw+ router queries the maximum frame size and adjusts as required. When the explorer response reaches the originator, the response indicates the maximum frame size supported on the entire path. For each explorer, DLSw+ adjusts the maximum frame size to be the minimum of its largest frame size (specified in the `dlsw local-peer` command), the largest frame size of the destination remote peer (specified in the `dlsw remote-peer` command and shared during the capabilities exchange), and the MTU on the local media. The default largest frame size used for remote peers varies by encapsulation type and is shown in Table 4-2. The default largest frame size in the `dlsw local-peer` command is 17,800.

Table 4-2 Default MTU and Largest Frame Sizes for Various Encapsulation Types and Media

Encapsulation Type	DLSw+ LF Default Values	IP/MTU Fragmentation	Orientation of WAN Transmission
TCP	17800	Yes	Byte stream
FST	516	No	Packet
Direct	MTU set on local interface	No	Packet

The largest frame size and the MTU interplay differently depending on the encapsulation type.

In general, when using TCP encapsulation, you probably do not need to change the largest frame size because TCP fragments the frame size according to the MTU. For example, if the LF is smaller than the MTU, then TCP fragments each packet and sends them in sections across the WAN. If the LF is larger than the MTU, then individual packets are placed into the TCP/IP frame.



If using FST encapsulation, you might need to change the largest frame size. You should change this value only if you know your traffic profile and your output WAN interface MTU and you need to increase throughput. For example, when using FST, the largest frame default is 516 to ensure that if the packet traverses Ethernet or serial interfaces, you do not exceed 1500 bytes when the DLSw, IP, and data-link control headers are added. If you know your traffic will not traverse an Ethernet LAN, you can increase the largest frame size. You should ensure that the length of the LAN Token Ring packet (less FCS) + 16 (DLSw header) + 20 (IP/FST header) does not exceed the MTU of any interface in the path. If the LF exceeds an MTU of an interface along the path, the packet is dropped and the session does not establish.

If using Direct encapsulation, you probably do not need to change the largest frame size because Direct encapsulation uses the MTU on the local interface to negotiate the LF size. As a result, the LF size never exceeds the MTU of an interface.

It is meaningful to increase the DLSw+ largest frame size only if the workstations can send larger frames. In this case, by allowing DLSw+ to send larger frames, you decrease the amount of segmentation required at the workstation. For example, if your message size is 1024 bytes and your maximum frame size on the path is 516 bytes, then the workstation needs to segment the frames. By setting the DLSw+ largest frame size to the next higher valid largest frame to accommodate a 1024-byte information field and all for protocol headers, then the workstation does not need to segment the message.

Set the largest frame size using the following `dlsw local-peer` command:

```
dlsw local-peer . . . [lf size]
```

where *size* can be one of the following amounts (bytes):

```
17800  
11454  
11407  
8144  
4472  
2052  
1500  
1470  
516
```

IP MTU Path Discovery

When IP MTU path discovery is configured, peering routers determine the maximum IP frame size to be used for the TCP peer connection during peer establishment. This maximum IP frame size then dictates the maximum number of SNA bytes that can be stored within one IP frame. The default size is 1450 bytes. Therefore, the maximum number of SNA bytes that can be stored within one IP frame is $1500 - (\text{TCP/IP header} + \text{data-link control}) = 1450$.

By increasing the maximum IP frame size, more SNA data can be placed within one TCP frame. This allows you to do the following:

- Increase WAN efficiency by sending large frames
- Decrease the number of TCP acknowledgments
- Reduce router CPU utilization

On the other hand, if bandwidth is not an issue, setting the IP frame size to a smaller number can improve response time.

By specifying IP MTU path discovery, when the peer session is established, each router along the path is queried for its MTU on the output interface. This is done by sending Internet Control Message Protocol (ICMP) echo packets of increasing sizes, with the don't fragment (DF) bit set. Intermediate routers that do not support that MTU size will respond with an "ICMP packet too big" message. Thus, the originating station knows when it has exceeded the MTU for that path (see RFC 1191 for more information).

Note: Setting all MTU sizes to larger values may impact the amount of memory used on the interface card. There is a limited amount of buffer space for the interface cards, and setting the MTU size higher on all interfaces may result in exhausting this memory. More memory is consumed by buffers if the MTU size is increased. On smaller platforms, such as the Cisco 2500 family of routers, this memory impact may be severe if you only have 2 MB of shared (I/O) memory.

The `ip tcp path-mtu-discovery` command is a global command not specific to an interface. When this command is active, the maximum IP frame size for a peer connection will be set to the minimum MTU path size on the path of that peer connection. Figure 4-6 and Figure 4-7 show how the packet size is affected when a network is configured with and without MTU path discovery.

Figure 4-6 DLSw+ Design without IP MTU Path Discovery

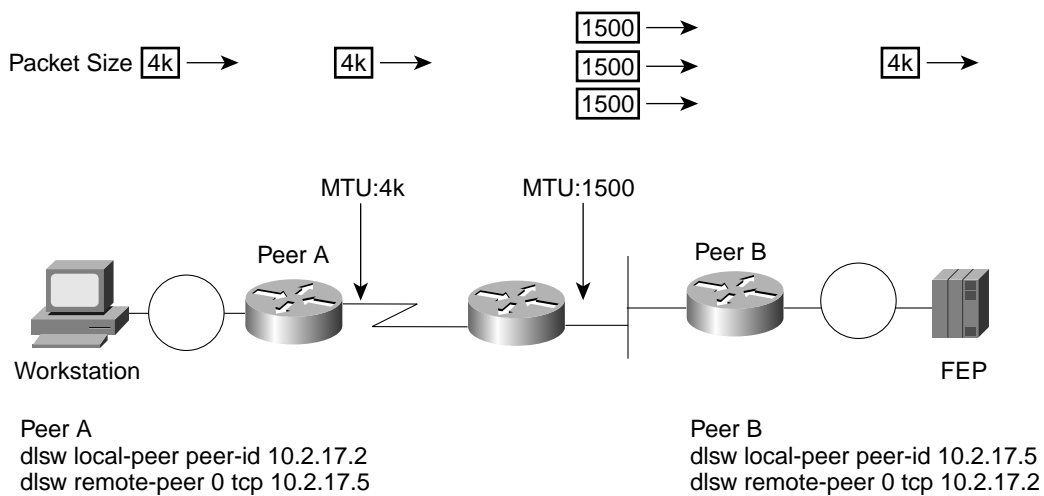
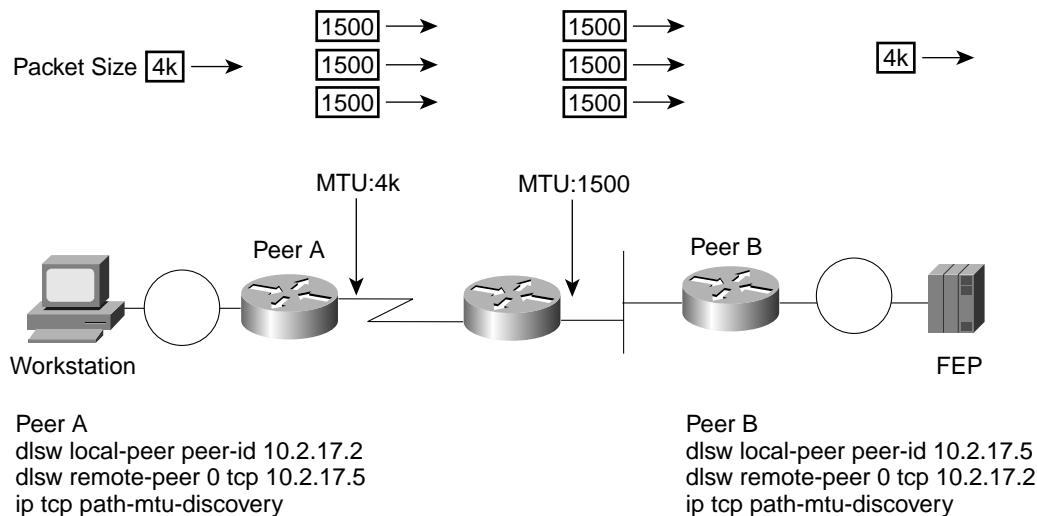


Figure 4-7 shows a DLSw+ network with IP MTU path discovery configured. IP MTU path prevents the packet from being fragmented at the IP layer. Fragmenting the packet at the IP layer causes problems because packets at this layer do not have the valuable TCP header packet identifier information, such as priority or destination. In Figure 4-7, the IP packet is automatically set to 1500 before being sent on the network, based on the minimum MTU size on the path of that peer connection. In Figure 4-6, however, the packet gets fragmented in route at the IP layer because the 4000 packet size is too large for the MTU set on the intermediate IP router.

Figure 4-7 DLSw+ Design with IP MTU Path Discovery



Packet assembly benefits from IP MTU path discovery because during the packet assembly process, more SNA frames can be stored within the TCP frame. For example, if 100 users in a remote location all require 3270 access to the central host, then all SNA request packets are destined for the same DLSw+ router. During heavy access periods, it is likely that many SNA requests will arrive at the remote router within a short period of time. These multiple SNA frames, all destined for the same host router, can be placed within the same TCP frame. When the TCP frame is successfully sent to the host router, one TCP acknowledgment can satisfy all the SNA requests.

This packet assembly only occurs during congestion when multiple SNA frames are in the queue. If there is no congestion, it is likely that one SNA packet will map to one TCP frame. DLSw+ does not wait for the multiple packets to arrive in the queue because this would impact end-user response time.

Note: When running DLSW+ over low-speed lines (4.8 or 9.6 kbps), an MTU of 576 provides more consistent response time. Use custom queuing to ensure that SNA gets three times the bandwidth of all other traffic so that an entire screen update is processed at one time.

Timer Settings

There are two types of timer settings: LLC2 timers and DLSw+ timers. The only timer discussed in this section is the LLC2 timers.

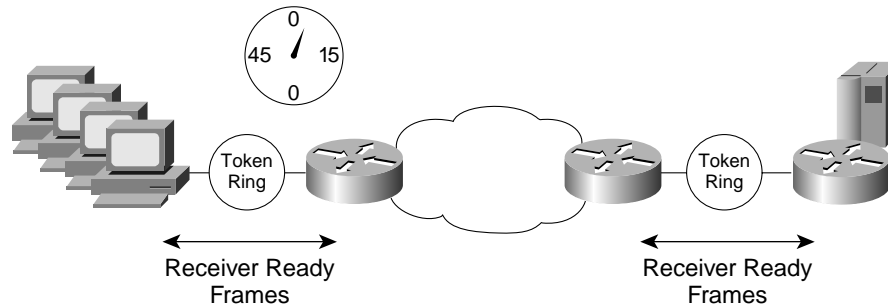
LLC2 Idle Time

LLC2 is a connection-oriented data-link control. Therefore, the end stations involved in the LLC2 connections must periodically check that the LLC2 connection is still active. One way of knowing that a connection is active is by sending and receiving I-frames over the LLC2 connection. Each frame requires an acknowledgment that not only indicates successful receipt of a frame, but also indicates that the connection is still alive. If there is a period of time when no I-frames (in other words, user data) traverse the LLC2 connection, then each workstation must send an LLC2 packet, a receiver ready, to its partner and receive a response to confirm that the LLC2 connection is still operational. The time that the end stations wait during idle traffic periods before sending a receiver ready frame is called the LLC2 idle time.

Every time the end station sends or receives a frame, it resets its LLC2 idle timer. If the idle timer expires, then the station sends an LLC2 packet to its partner. If there are many thousands of LLC2 sessions, then many LLC2 receiver ready messages traverse the network during idle periods of time.

When a router is locally terminating the LLC2 session, as shown in Figure 4-8, it is the responsibility of the router to adhere to the LLC2 protocol. Thus, during periods of inactivity, the router must send LLC2 requests or acknowledge LLC2 requests from the workstations. This can place an unnecessary load on the router, which can be avoided by increasing the LLC2 idle timer parameter on the LAN segment.

Figure 4-8 LLC2 Receiver Ready Messages Flowing Between End Systems and DLSw+ Routers (One LLC2 Connection at Each DLSw+ Router for Every SNA PU or NetBIOS Session)



A larger LLC2 idle timer value should be implemented when there is a large number of LLC2 sessions. Increasing the LLC2 idle time when supporting 4000 LLC2 sessions decreases the router CPU utilization significantly. The trade-off is that it takes longer to identify time-out conditions. This condition is generally a good trade-off.

A value of 30,000 milliseconds (30 seconds) is suggested, although LLC2 idle time can be increased to as much as 60,000 milliseconds (60 seconds). Use the following syntax to configure this command:

```
llc2 idle-time milliseconds
dlsw bridge-group idle-time milliseconds
```

The `dlsw bridge-group idle-time` command affects the LLC2 idle time on a transparent bridge interface. Although it is possible to configure the LLC2 idle time on an Ethernet interface, the timer is not affected. The LLC2 timer does affect the Ethernet interface, however, when the DLSw+ Ethernet Redundancy feature is enabled because transparent bridging is not configured.

The maximum value is 60,000. The command to set the LLC2 idle timer is an interface subcommand. Apply it to the appropriate LAN segment. A sample configuration follows:

```
source-bridge ring-group 100
dlsw local-peer peer-id 172.26.1.1
dlsw remote-peer 0 tcp 172.26.10.1
interface token-ring 0
ip address 172.26.1.1 255.255.255.0
 source-bridge 3 1 100
 llc2 idle-time 30000
. . .
```

DLSw+ Timers

There are several timers in DLSw+ that you can set with a `dlsw timer` command. In general, you do not need to modify these timers. A description of them is included here for completeness along with considerations on the impact of changing them. To change timers, use the following command:

```
dlsw timer {timer-type} time
```



Where *time* is specified in seconds or minutes and *timer-type* can be any of the following keywords:

- explorer-delay-time
- icannotreach-block-time
- netbios-cache-timeout
- netbios-explorer-timeout
- netbios-group-cache
- netbios-retry-interval
- netbios-verify-interval
- sna-cache-timeout
- sna-explorer-timeout
- sna-group-cache
- sna-retry-interval
- sna-verify-interval
- explorer-wait-time

The explorer-delay-time is the time the router waits before responding to explorers. Use this option on the router that is load balancing traffic. Because the DLSw+ peer selects its new circuit paths from within its reachability cache, the peer performing the load balancing needs enough time to receive all the explorer responses. The valid range is 1 to 5 minutes. It defaults to 0.

The icannotreach-block-time is the time the router marks a resource unreachable after failing in an attempt to find it. While the resource is marked unreachable, searches for that resource are blocked. It is disabled by default. Use this option only if you have excessive explorer traffic and you want to avoid broadcasts for frequently accessed resources that are not currently available or are remote. If used, specify an amount of time that the user is willing to wait for a resource to recover. In some cases (typically in large NetBIOS networks), the NetBIOS station may be up and available, but because of traffic loads, the response may not come back in time. This may cause a peer to consider the station not reachable. If this timer is not specified or set to 0, the user can connect by retrying the command. If the timer is set to 10 minutes, the user cannot connect for 10 minutes.

The netbios-cache-timeout is the time that DLSw+ caches a NetBIOS name location for both the local and remote reachability caches. It defaults to 16 minutes. Setting it lower may cause more broadcasts. Setting it higher increases the chance of having an invalid cache entry. However, for frequently accessed resources, the router generally deletes an invalid cache entry before 16 minutes elapses, so setting this timer to a shorter period of time is probably not necessary.

Cache entries resulting from statically defined reachability paths are never deleted. Cache entries configured using the dlswn canreach command and learned as part of a capabilities exchange are deleted when the associated peer connection is taken down.

The netbios-explorer-timeout is the length of time that this router sends explorers to a NetBIOS resource (for LAN resources) or the time DLSw+ waits for a response before deleting the pending record (for remote resources). It defaults to 6 seconds. This timer has no impact on when a resource is marked unreachable. Its impact on the LAN is to determine how many retries are sent.

The netbios-group-cache is the length of time NetBIOS entries stays in the group cache. Use this option only on routers configured to be border peers. The valid range is 1 to 86000 seconds. It defaults to 240 seconds (4 minutes).

The netbios-retry-interval is the interval DLSw+ waits for a response to a name query or add name query on a LAN before retransmitting the request. The default is 1 second. Retries continue to be sent until the NetBIOS explorer timeout is reached (retries are not sent across the WAN).

The `netbios-verify-interval` is the interval between the creation of a cache entry and when the entry is marked stale. If a cache entry is marked stale and a search request comes in for that entry, a directed verify is sent to ensure it still exists. A directed verify is an explorer (for example, NetBIOS NAME-QUERY) sent directly to each cached peer (on the WAN) or a single route explorer sent over every port in the cache (on the LAN). The default is 4 minutes. Setting this value higher increases the time it takes for a resource to be found if its cached location is invalid.

The `sna-cache-timeout` is the length of time that DLSw+ caches the MAC or SAP of an SNA resource before it is discarded. It defaults to 16 minutes. Setting the timer lower may cause more broadcasts. Setting it higher increases the chance of having an invalid cache entry. However, for frequently accessed resources, the router generally deletes an invalid cache entry before 16 minutes elapse, so setting this timer to a shorter period is probably not necessary.

Cache entries resulting from statically defined reachability paths are never deleted. Cache entries configured using the `dls w icanreach` command and learned as part of a capabilities exchange are deleted when the associated peer connection is taken down.

The `sna-explorer-timeout` is the length of time that this router sends explorers to a NetBIOS (for LAN resources) or the time DLSw+ waits for a response before deleting the pending record (for remote resources). It defaults to 3 minutes. This timer has no impact on when a resource is marked unreachable. Its impact on the LAN is to determine how many retries are sent. When using either FST or direct encapsulation without local acknowledgment, this frame is sent over an unreliable mechanism, so it is possible for high volumes of traffic to cause frame drops. In this case, you may want to configure a smaller value for this timer to shorten the time it takes to find resources.

The `sna-group-cache` is the length of time SNA entries stay in the group cache. Use this option only on routers configured to be border peers. The valid range is 1 to 86000 seconds. It defaults to 240 seconds (4 minutes).

The `sna-retry-interval` is the interval DLSw+ waits for a response to a TEST or XID request on a LAN before retransmitting the request. The default is 30 seconds.

The `sna-verify-interval` is the interval between the creation of a cache entry and when the entry is marked stale. If a cache entry is marked stale and a search request comes in for that entry, a directed verify is sent to ensure it still exists. A directed verify is a CANUREACH frame sent directly to every cached peer (on the WAN) or a single route explorer sent over every port in the cache (on the LAN). The default is 4 minutes. Setting this value higher increases the time it takes for a resource to be found if its cached location is invalid.

The `explorer-wait-time` is the number in seconds that DLSw+ waits after sending an explorer before picking a peer as the best path. When DLSw+ starts exploring, it waits for *time* seconds before responding to the TEST frame. Setting this timer to 1 to 2 seconds gives DLSw+ time to learn all possible peers before selecting the least-cost peer. Do not modify this timer unless you have multiple central site peers, you are using cost to select a preferred peer, and your capable peer frequently responds first before your preferred peer.

When configuring the Ethernet Redundancy feature, you can use the `dls w transparent timers` command to set the timeout value that the master router waits for all requests for a circuit before giving permission to a router to take a circuit. You can create separate timeout values for SNA and NetBIOS sessions. The default NetBIOS value is 400 milliseconds and the default SNA value is 1000 milliseconds. It is not one of the options in the `dls w timer` command, rather it is a separate command.

Queue Depths

During congestion, packets might get queued in the router. You can control the depth of certain queues to improve network performance.

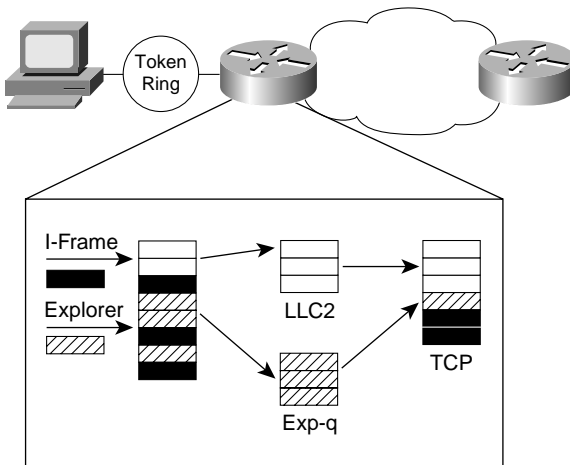
Explorer Queue Depth

Explorers are used to find resources in DLSw+ and on LANs. Explorer caching by DLSw+ helps decrease the steady state explorer load on both the network and on DLSw+ routers. When a DLSw+ router receives an explorer for a cached resource, it either responds locally or sends a directed explorer.

Problems occur when there is an excessive amount of broadcast traffic (known as a broadcast storm) and the explorers arrive at a rate faster than DLSw+ can process them. To address this, you can use the source-bridge explorerq-depth command. By using this command, you limit the number of explorers that can be queued waiting to be processed. Without doing this, a broadcast storm may cause input buffers to fill up with explorer traffic, preventing end-user traffic from getting through. Dropping these excessive explorers also minimizes CPU utilization.

Figure 4-9 illustrates explorer processing. When an explorer queue is full, any incoming explorers are dropped, causing end systems to retransmit the explorers. By limiting the size of the SRB explorer queue, you can ensure that explorer traffic does not monopolize DLSw+ buffers.

Figure 4-9 Explorer Processing in a DLSw+ Router



The syntax of the command is:

```
source-bridge explorerq-depth depth
```

Where *depth* is the maximum number of incoming explorers. When this number is reached, new explorers are dropped. If you have excessive explorer traffic, set this value to between 10 and 20.

Typically, when there is an explorer storm, most explorers are destined to the same MAC address. Dropping these explorers (when the queue is full) gives the router time to receive the reply to the explorers that were processed and, therefore, obtain a cache hit. When the cache hit is obtained, the router can often respond to the explorers without forwarding them.

Input Hold Queue

This queue is used to keep track of input frames off the LAN segment (other interface types as well, but here we will concentrate on LAN segments) that are awaiting system processing. During peak loads, you may see some buildup (or drops) in this queue. (Use the show interface command to get this information. See the chapter “Using

Show and Debug Commands” for more information.) Some protocols that are very traffic intensive during startup may require the input hold queue to be increased. Increasing the hold queue enables the router to simultaneously process more packets from a particular interface.

A good example of this is a startup of APPN sessions. There are many small packets that flow during startup, and it is not unusual to see a buildup in the input hold queue (in other words, the packets come off the Token Ring segment much faster than the router can process them out the WAN ports).

It should be noted that if you see constant drops on the input hold queue, then increasing the input hold queue does not help. There is probably another problem in the network. Increasing the input hold queue can help when there is a transient load (for example, at startup) where the router needs the ability to hold on to a few more packets than normal. This alleviates packet retransmission and minimizes the possibility of further dropped packets.

This command is an interface subcommand. It can be applied to any interface. The syntax of the command is:

```
hold-queue length in
```

Where *length* is the number of buffers that can be stored. The default is 75 input buffers. The following is a sample configuration:

```
source-bridge ring-group 100
dlsw local-peer peer-id 172.26.1.1
dlsw remote-peer 0 tcp 172.26.10.1
interface token-ring 0
 ip address 172.26.1.1 255.255.255.0
 source-bridge 3 1 100
 llc2 idle-time 30000
 hold-queue 200 in
```

System Buffers

In an SNA environment, dropping system buffers is not good. Consistently dropping buffers leads to SNA session loss, and therefore, system buffer tuning is required to prevent this situation.


Note: This section describes how to diagnose a system buffer problem and to compile enough information so that a Cisco engineer (system engineer or customer engineer) can assist with the system buffer changes. Do not attempt to adjust buffers without assistance.

System buffers come in various sizes (small, middle, large, very large, and huge). The memory used for these buffers is called I/O or shared memory. Low-end routers have one memory location for I/O and another memory location for main memory. High-end routers have one block of memory split into main and I/O.

For process switched traffic, when a packet arrives in the router, it is placed in the smallest size buffer that can accommodate it. If that size buffer is not available and the router can create another buffer quickly enough, it does. When this new buffer is created, it stays in the pool temporarily but is trimmed back later. This freed memory can then be used to create any other size buffer.

If the router cannot create a buffer in time, a buffer miss is recorded. If the router cannot create a buffer because there is no more I/O memory available, then a no memory condition is recorded. Not having enough I/O memory available indicates a problem.

Two show commands are used to diagnose buffer problems: show memory and show buffers. The show memory command displays the total amount of memory available, memory used, and memory currently available. The show buffers command details all the buffer information: number of misses, number of no memory conditions, and number of buffers assigned.



If you suspect a memory problem, check the status of your buffers using the `show buffers` command. If you see some buffer misses, do not be alarmed. It is not unusual to see some misses (in other words, if the router has been running for several weeks, you may see that over this time you have 100 misses).

If you view your buffers and see that the miss count is incrementing (by issuing a few `show buffers` commands), then take note of which buffer size is being missed.

When you have the details of the buffer misses, issue the `show memory` command and take note of the amount of shared (or I/O) memory that is still available. If this value is still larger than 1 MB, it is likely that tuning your buffers will alleviate the buffer misses.

If you note that no memory conditions are occurring (from the `show buffers` command), note the amount of free shared (or I/O) memory (from the `show memory` command). If you find that the amount of free shared memory is almost zero, it is a serious condition. This occurs for one of two reasons: either the router needs more I/O memory to accommodate the amount of traffic and flow control requirements, or you have tuned your buffers and over-allocated in some area and depleted the I/O memory.

When you have gathered this information, open a case with the Cisco Technical Assistance Center (TAC), or discuss it with your systems engineer. You should supply the following information:

- Current configuration (issue a write terminal command to get this information)
- Description of the symptom (for example, session drops, poor response time, and so forth)
- Output of `show memory` command (but typically not the whole memory map, just the initial information)
- Output of `show buffers` command (you may want to include the output from multiple `show buffers` commands if you are trying to show an increase in buffer misses. Be sure to track how much time was allowed to lapse between outputs.
- The current Cisco IOS release you are using, which you can determine by issuing a `show version` command

Miscellaneous Customization Options

SRB Explorers

By default, when Cisco's DLSw+ initiates an explorer, it sends a single route explorer. Most SRB implementations respond to a single route explorer with an all routes explorer so that the best possible path can be selected. If you have an implementation that does not respond to single route explorer with an all routes explorer, you can configure DLSw+ to send explorers as all routes explorers using either the `dlsw allroute-sna` or the `dlsw allroute-netbios` command.

Initial and Maximum Pacing Windows

DLSw+ uses an adaptive pacing flow-control algorithm that automatically adjusts to congestion levels in the network. (This algorithm is described in the "Introduction.") The default initial pacing window size is 20 and the default maximum pacing window size is 50. Some environments need the ability to adjust this window size. The capability to modify the default window sizes was added in Cisco IOS Release 10.3(14), 11.0(11), 11.1(5), and 11.2.

You may want to set the initial pacing window to a lower value if one side of a connection can send far more data than the other side can receive, for example, if you have a Frame Relay network and the central site router accesses over a T1 link and the remote router accesses over a 56-kbps link. With Cisco IOS Release 11.2, Committed Information Rate (CIR) enforcement provides an alternate way to address this issue. You may want to set the initial or maximum pacing sizes to higher values if one side is frequently waiting for permission to send more traffic and the other side is capable of handling more traffic.

To determine if you should modify either of these defaults, you can use the `show dlsw circuits` command, which shows the current window packets and permitted and granted packets. If the current window shows the maximum of 50 and the permitted and granted packets shows 0 for some time, this indicates that the adaptive pacing has increased to the maximum, but one side is still frequently waiting before it can send more. In this case, you may improve your throughput by increasing the maximum pacing window.

If the current window packet is higher than the initial pacing window but less than the maximum pacing window, and the permitted and granted packet is 0 or very small, it may be a signal that the adaptive pacing algorithm is increasing the window size but is not increasing the window size quickly enough. In this case, you may improve your throughput by increasing the initial pacing window.

If the current window packet is less than the initial pacing window, it may indicate that the receiver cannot absorb traffic as quickly as it can be sent. In this case, you may want to reduce the initial pacing window.

To modify these pacing values, include the following keywords on the `dlsw local-peer` command:

```
dlsw local-peer . . . [init_pacing_window size] [max_pacing_window size]
```

Where *size* can be anything between 1 and 50, but `max_pacing_window` should always be larger than `init_pacing_window`.