Teldat Router

Bridge

Doc. *D717-I* Rev. *10.00*
April, 2003

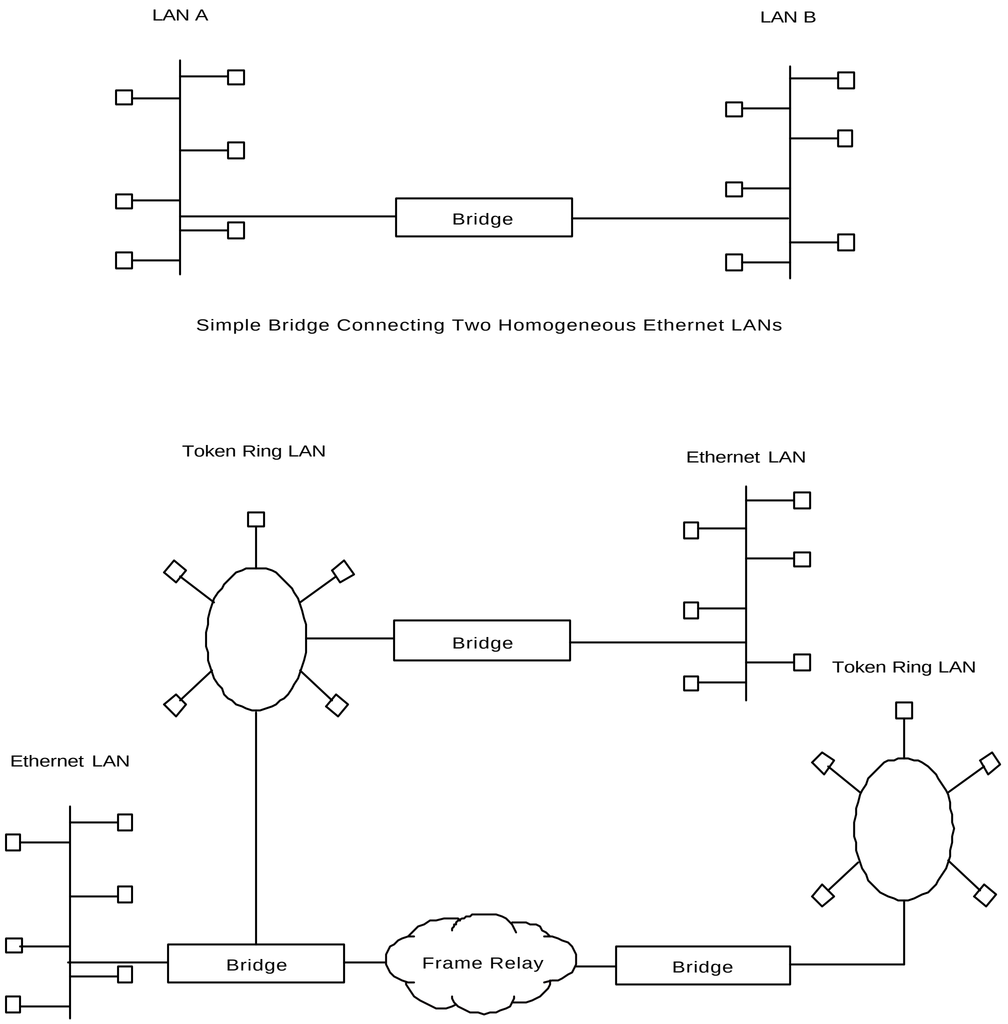
INDEX

Chapter 1
Fundamentals of Bridging

1. About Bridges

A bridge is a device that links two or more *Local Area Networks* (LANs). The bridge accepts data frames from each connected network and then decides whether to forward each frame based on the *Medium Access Control (MAC)* frame.

You can use *bridges* to link homogeneous or heterogeneous networks. The term homogeneous means that the connected networks use the same *bridging* method and media types. The term heterogeneous means that the connected networks mix different *bridging* methods and media types, and offer more configuration options.



Simple Bridge Connecting Two Homogeneous Ethernet LANs

Complex Bridge Connecting Heterogeneous LANs

Figure 1-1. Homogeneous and Heterogeneous Bridging Configurations

2. Bridges and Routers

Bridges and *routers* connect network segments. However, each device uses a different method to establish and maintain the *LAN* to *LAN* connections. Routers connect LANs at layer 3 (network layer) of the *OSI* model while *bridges* connect LANs at layer 2 (data link layer).

2.1. Router Connections

Routers connect distant and diverse LANs more intelligently using network layer protocols. Because of the in-depth network topology related information available at network layer, using routers to connect large networks is recommended.

You must route when a protocol is routable. For example, you must route when mixing Ethernet and Token Ring with protocols that use MAC information in the upper layers

2.2. Bridge Connections

Bridges connect LANs across a physical link. This connection is essentially transparent to the host connected on the network.

A bridge acts as a relay for frames between networks at the data link layer. The data link layer maintains physical addressing schemes, line discipline, topology reporting, error notification, flow control, and ordered delivery of data frames. The principal service provided by the data link layer to the higher layer is that of error detection and control. With a fully functional data link layer protocol, the next higher layer may assume virtually error-free transmission over the link.

You must bridge when the protocol is non-routable, that is, it carries no network layer.

2.3. Advantages of Bridging

Isolation from upper-layer protocols is one of the advantages of bridging. Since bridges function at the data link layer, they are not concerned with looking at the protocol information that occurs at the upper layers. This provides for lower processing overheads and fast communication of network layer protocol traffic.

Bridges can also filter frames based on layer 2 fields. This means that the bridge may be configured to accept and forward only frames of a certain type or ones that originate from a particular network. This ability to configure filters is very useful for maintaining effective traffic flow.

Bridges are advantageous when dividing large networks into manageable segments. The advantages of bridging in large networks can be summed up as follows:

- Bridging lets you isolate specific network areas, giving them less exposure to major network problems.
- Filtering lets you regulate the amount of traffic that is forwarded to specific segments.
- Bridges allow communication between more internetworking devices than would be supported on any single LAN connected to a bridge.
- Bridging eliminates node limitation. Local network traffic is not passed on to all of the other connected networks.
- Bridges extend the connected length of a LAN by allowing the connection of distant workstations.

2.4. Bridging Interfaces

Bridging interfaces include combinations of one or more of the following:

- Ethernet
- Token Ring
- Serial line (where data link is Serial, PPP, and Frame Relay)

The Ethernet interfaces support transparent bridging.

The Token Ring interface supports source routing and transparent bridging.

The serial line interface provides point-to-point connectivity for transparent and source routing traffic. It is important to note that a bridge configuration over a serial line should be consistent at both endpoints. This means that you must configure both endpoints as follows:

- Transparent to transparent
- Source routing to source routing
- Source routing/transparent to source routing/transparent

It is best if the serial line is configured for both bridging methods if you want mixed bridging. Make sure that bridging routers are consistent in their bridging method or in their routing of particular protocols.

3. Bridges methods

Bridging is comprised of two pure protocols or methodologies: Source Transparent Bridging (STB), and Source Route Bridging (SRB).

- STB is a bridging method primarily for Ethernet environments in which bridges automatically develop bridging tables and update those tables in response to a changing topology.
- SRB is a bridging method solely for Token Ring environments in which the sending station determines the route that the frame will follow and includes the routing information, or path, that is built by routers participating in SRB.

You can use STB and SRB alone or in combination to meet your requirements regardless of media or network topology. These combinations are Source Route Transparent Bridging (SRT), Source Route-Transparent Bridging (SR-TB Conversion), and Adaptive Source Route Transparent Bridging (ASRT).

- SRT is a method of bridging both source routing frames and transparent frames based on the Route Information Indicator (RII). It can be thought of as two bridges in one.
- SR-TB is a method of bridging between SRB domains and STB domains. It does this through a conversion process between the two bridging technologies (IBM 8209).
- ASRT is TELDAT’s enhancement to SRT bridging technology. It combines SRT and SR-TB functionality. It allows all end stations in a complex bridged environment to communicate without the standard limitations. Tables are maintained for SRB and STB end stations so that they can be bridged or converted as required.

Note: SRT and ASRT both require Content Addressable Memory (CAMs) in order to bridge transparently over Token Ring in CNX platforms.

The decision to choose one method of bridging over another depends on the network’s topology and the applications used on the end stations.

4. How Bridges Work

Bridges function at the MAC level. According to the IEEE 802 LAN standard, all station addresses are specified at the MAC level. The following examples show how a bridge functions at the MAC level.

4.1. Example 1: Local Bridge Connecting Two LANs

Figure 1-2 shows a two-port bridge model connecting end stations on two separate LANs. In this example, the local bridge connects LANs with identical LLC and MAC layers (i.e. two Token Ring LANs).

The bridge captures MAC frames whose destination addresses are not on the local LAN and forwards them to the appropriate destination LAN. Throughout this process, there is a dialogue between the peer LLC entities in the two end stations. Architecturally, the bridge need not contain an LLC layer since the function of the LLC layer is merely to relay MAC frames.

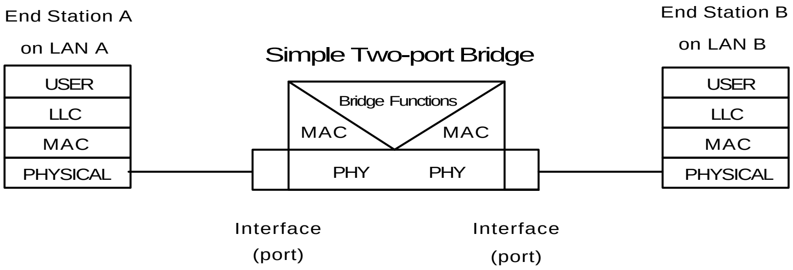


Figure 1-2. Two-port Bridge Connecting Two LANs

4.2. Example 2: Remote Bridging over a Serial Link

Figure 1-3 shows a pair of *bridges* connected over a serial link. These remote bridges connect LANs with identical LLC and MAC layers (i.e. two Token Ring LANs).

Bridge A captures a MAC frame whose destination address is not on the local LAN and then sends it to bridge B across a serial line using the appropriate serial line encapsulation to identify the bridge frame type. Remote bridge B decapsulates the serial line header and forwards the frame to the local LANs. Throughout this process, there is a dialogue between the peer LLC entities in the two end stations.

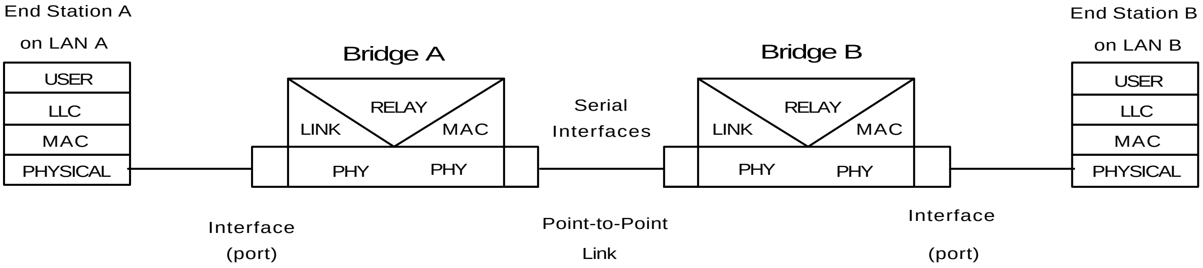


Figure 1-3. Bridging Over a Point-to-Point Link

Data is encapsulated as the bridges communicate data over the serial link.

Figure 1-4 illustrates the encapsulation process.

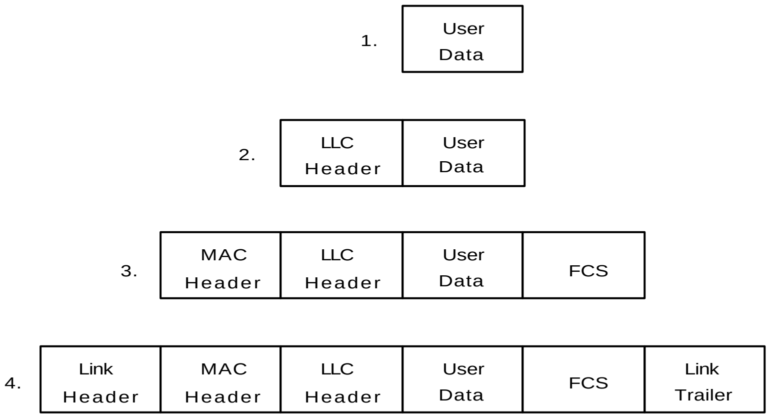


Figure 1-4. Data Encapsulation Over a Point-to-Point Link

Encapsulation proceeds as follows:

1. End station A provides data to its LLC.
2. LLC appends a header and passes the resulting data unit to the MAC level.
3. MAC then appends a header and trailer to form a MAC frame. Bridge A captures the frame.
4. Bridge A does not strip off the MAC fields because its function is to relay the intact MAC frame to the destination LAN. In the point-to-point configuration, however, the bridge appends a link layer (e.g. HDLC) header and trailer and transmits the MAC frame across the link.

When the data frame reaches Bridge B (the target bridge), the link fields are stripped off and Bridge B transmits the *original, unchanged* MAC frame to its destination, end station B.

4.3. MAC Bridge Frame Formats

As mentioned, bridges interconnect LANs by relaying data frames between the separate MAC entities of the bridged LANs. MAC frames provide the necessary forwarding information in the form of source and destination addresses. This information is essential for the successful transmission and reception of data.

IEEE 802 supports three types of MAC frames:

- CSMA/CD (802.3)
- Token bus (802.4)
- Token Ring (802.5)

> ***Note: A separate frame format is used at the LLC level. This frame is then embedded in the appropriate MAC frame.***

Figure 1-5 shows the CSMA/CD and Token Ring MAC frame formats supported by the bridges. The specific frames are detailed in the following section.

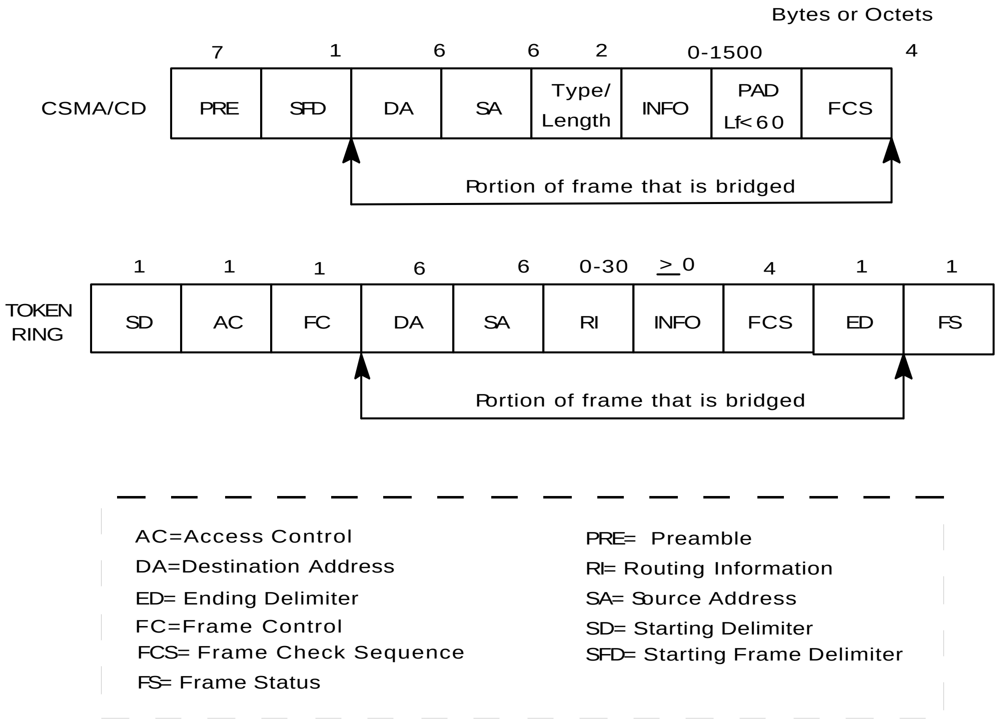


Figure 1-5. MAC Frame Format Samples

4.4. CSMA/CD (Ethernet) MAC Frames

The following information describes each of the fields found in CSMA/CD (Ethernet) MAC frames:

- *Preamble (PRE)*. 7-byte pattern used by the receiving end station to establish bit synchronization and then locate the first bit of the frame.
- *Start Frame Delimiter (SDF)*. Indicates the start of the frame.

The portion of the frame that is actually bridged consists of the following fields:

- *Destination Address (DA)*. Specifies the end station for which the frame is intended. This address may be a unique physical address (one destination), a multicast address (a group of end stations as a destination). The format is 48-bit (6 octets) and must be the same for all stations on that particular LAN.
- *Source Address (SA)*. Specifies the end station that transmitted the frame. The form must be the same as the destination address format. This address must never be multicast or broadcast address.
- *Type/Length*. Specifies the number of LLC bytes that follow. If the value of this field is less than 0x600 then the field is the length value of the LLC bytes that follow. These are normally known as IEEE 802.3 frames. The value greater or equal to 0x600 identifies the protocol type. This is known as Ethernet-II frame.
- *Info (INFO)*. Embedded fields created at the LLC level that contain service access point information, control information, and user data.
- *Pad*. Sequence of bytes that ensures that the frame is long enough for proper collision detection (CD) operation. The minimum frame size on Ethernet is 60 bytes excluding FCS.
- *Frame Check Sequence (FCS)*. 32-bit cyclic redundancy check value. This value is based on all fields, starting with the destination address.

4.5. Token Ring MAC Frames

The following information describes each of the fields found in Token Ring MAC frames:

- *Starting Delimiter (SD).* Unique 8-bit pattern that indicates the start of the frame.
- *Access Control (AC).* Field with the form at PPPTMRRR where PPP and RRR are 3-bit priority and reservation variables, M is the monitor bit, and T indicates that this is either a Token or data frame. If it is a Token, the only other field is the ending delimiter (ED).
- *Frame Control (FC).* Indicates if this is an LLC data frame. If not, bits in this field control operation of the Token Ring MAC protocol.

The portion of the frame that is actually bridged consists of the following fields:

- *Destination Address (DA).* Same as CSMA/CD and token bus, except that bit format is non-canonical.
- *Source Address (SA).* Identifies the specific station that originates the frame. The length of the field can be a 6 octet address. The bit format is non-canonical.
- *Routing Information Field (RIF).* When the RII (most significant bit of most significant byte) in the source address field is set to 1, this field appears after the source address. The RIF is required for the source routing protocol. It consists of a 2-octet routing control field and a series of 2-octet route designator fields.
- *Info (INFO).* Embedded fields created at the LLC level containing service access point information, control information, and user data.
- *Frame Check Sequence (FCS)* A 32-bit cyclic redundancy check value. This value is based on all fields, starting with the destination address.

Finally, the *End Delimited (ED)* contains the error detection (E) bit, and the intermediate frame (I) bit. The I bit indicates that this is the frame other then the final one of a multiple *frame* transmission. The *Frame Status (FS)* contains the address recognized (A) and frame copied (C) bits.

4.6. Pseudo-serial Ethernet Frames

Pseudo-serial Ethernet is an optional mode of operation. It provides for the encapsulation of any routed protocol on a bridging router proprietary serial line so that it can be forwarded within an Ethernet encapsulated frame. This allows the protocol to communicate with a pure bridge on the opposite end of the serial line.

This mode makes the serial lines appear as an Ethernet interface to the configured routing protocols. The handler uses Ethernet (or IEE 802.3 as appropriate) encapsulations, thus limiting the protocols to the maximum Ethernet frame size. These Ethernet frames are then sent and received as *bridged* Ethernet frames on the serial line. Any frames arriving on the routed protocol code points from the serial lines are ignored, and bridge Ethernet frames will be passed to the bridge or routing forwarders as appropriate.

This encapsulation is normally not necessary with the bridging routers at both ends of the serial line, since both can be configured to route the same set of protocols over the same serial line.

Chapter 2
Using Transparent Bridging (STB)

1. About STB

The Transparent Bridge is also commonly known as a Spanning Tree Bridge (STB). The term transparent refers to the fact that the bridge silently forwards non-local traffic to attached LANs in a way that is transparent or unseen to the user. End station applications do not know about the presence of the bridge. The bridge learns about the presence of end stations by listening to traffic passing by. From this listening process it builds a database of end station addresses attached to its LANs.

For each frame it receives, the bridge checks the frame’s destination address against the ones in its database. If the destination is on the same LAN, it does not forward the frame. If the destination is on another LAN, it does forward the frame. If the destination address is not present in the database, it forwards the frame to all the LANs connected to the bridge except the LAN from which it originated.

All transparent bridges use the spanning tree protocol and algorithm. The spanning tree algorithm produces and maintains a loop-free topology in a bridged network that may contain loops in its physical design. In a mesh topology, where more than one bridge is connected between two LANs, data packets can bounce back and forth between two LANs’ parallel bridges. This creates a redundancy in data traffic and produces the phenomenon known as looping.

Without spanning tree, when looping occurs, you must configure the local and/or remote LAN to remove the physical loop. With spanning tree, a self-configuring algorithm allows a bridge to be added anywhere in the LAN without creating loops. When you add the new bridge, the spanning tree transparently reconfigures all bridges on the LAN into a single loop-free spanning tree.

Spanning tree never has more than one active data route between two end stations, thus eliminating data loops. For each bridge, the algorithm determines which bridge ports to use to forward data and which ones to block to form a loop-free topology. Among its features spanning tree provides the following:

- *Loop detection.* Detects and eliminates physical data link loops in extended LAN configurations.
- *Automatic backup of data paths.* Deliberately configured from redundant paths. The bridges connecting to the redundant paths enter backup mode automatically. When a primary bridge fails, a backup bridge becomes active.
- *User configurability.* Lets you tailor your network topology. Sometimes the default settings do not produce the desired network topology. You can adjust the bridge priority, port priority and path cost parameters to shape the spanning tree to your network topology.
- *Seamless interoperability.* Allows LAN interoperability without configuration limitations caused by diverse communications environments.
- *Bridging of non-routing protocols.* Provides cost-effective bridging of non-routing protocols such as Digital Equipment Corporation’s Local Area Transport (LAT) terminal protocol.

2. Routers and STB

When bridge and router software run concurrently on a router equipped with the spanning tree option, the following occurs:

- Packets are routed if a specific protocol forwarder is globally enabled.
- Packets are filtered if you configure specific protocol filters.
- Packets that are not routed or filtered are candidates for bridging depending on the destination MAC (Medium Access Control) address.

3. STB Network Requirements

The transparent bridge implements a spanning tree bridge that conforms to the IEEE 802.ID standard. All transparent bridge such as Ethernet, SL, and TKR on the network must be 802.ID spanning tree bridges. This Spanning Tree protocol is not compatible with bridges implementing the proprietary Digital Equipment Corporation spanning tree protocol used in some older bridges.

4. Enabling STB

The following information outlines the initial steps required to enable the transparent bridging option offered by the ASRT bridge.

Note: Transparent bridging over X.25 is not supported. You can work around this by configuring the IP tunnel feature.

Use the following commands to enable transparent bridging:

- ***Bridge***. Enables transparent bridging.
- ***Port*** #. Enables bridging for a determined interface. Execute this command for all LAN/WAN interfaces over which the bridge is going to operate.
- ***No transparent port#***. Disables transparent bridging on previously enabled bridge port. Repeat the command for all ports you want excluded from the transparent bridging configuration.

After completing the procedures just described, you can enter **LIST BRIDGE** to check your configuration.

To make changes to the configuration, see **Chapter 7 "ASRT Configuration"** of this manual. After you finish making the changes to the configuration, you must restart the router for the new configuration to take effect.

5. How STB Works

During startup, all participating bridges in the network exchange Hello Bridge Protocol Data Units (BPDUs), which provide configuration information about each bridge. BPDUs include information such as the bridge ID, root ID, and root path cost. This information helps the bridges to determine unanimously which bridge is the *root bridge* and which bridges are the designated bridges for LANs to which they are connected.

Of the information exchanged in the Hello messages, the following parameters are the most important for computing the spanning tree:

- *Root bridge ID*. The bridge ID of the root bridge, the designated bridge for all the LANs to which it is connected.
- *Root path cost*. The sum of the designated path costs to the root via this bridge's root port. This information is transmitted by both the root bridge and the designated bridges to update all bridges on path information if the topology changes.
- *Bridge ID*. A unique ID used by the spanning tree algorithm to determine the spanning tree. Each bridge in the network is assigned a unique bridge identifier.
- *Port ID*. The ID of the port from which the current Hello BPDU message was transmitted.

With this information available, the spanning tree begins to determine its shape and direction and then creates a logical path configuration as follows:

1. A root bridge for the network is selected by comparing the bridge Ids of each bridge in the network. The bridge with the lowest ID value (i.e. highest priority) wins.
2. The spanning tree algorithm then selects a designated bridge for each LAN. If more than one bridge is connected to the same LAN, the bridge with the smallest path cost to the root is selected as the designated bridge. In the case of duplicate path costs, the bridge with the lowest bridge ID is selected as the designated bridge.
3. The non-designated bridges on the LANs put each port that has not been selected as a root port into a *blocked* state. In the *blocked* state a bridge still listens to Hello BPDUs so that it can act on any changes that are made in the network (e.g. designated bridge fails) and change its state from blocked to forwarding (i.e. forwarding data).

Through this process, the spanning tree algorithm reduces a bridged LAN network of arbitrary topology into a single spanning tree. With the spanning tree there is never more than one active data path between any two end stations, thus eliminating data loops.

This new configuration is bounded by a time factor. If a designated bridge fails or is physically removed, other bridges on the LAN detect the situation when they do not receive Hello BPDUs within the time period set by the bridge maximum age time. This event triggers a new configuration process where another bridge is selected as the designated bridge. A new configuration is also created if the root bridge fails.

6. Shaping the Spanning Tree

When the spanning tree uses its default settings, the spanning tree algorithm generally provides acceptable results. The algorithm may, however, sometimes produce a spanning tree with poor network performance. In this case you can adjust the bridge priority, port priority, and path cost to shape the spanning tree to meet your network performance expectations. The following example as shown in Figure 2 - 1 explains how to do this.

Figure 2 - 1 shows three LANs networked using three bridges. Each bridge is using default bridge priority settings for its spanning tree configuration. In this case, the bridge with the lowest physical address is chosen as the root bridge since the bridge priority of each bridge is the same. In this example, this is Bridge 2.

The newly-configured spanning tree stays intact due to the repeated transmissions of Hello BPDUs from the root bridge at a present interval (bridge Hello time). Through this process, designated bridges are updated with all configuration information. The designated bridges then regenerate the information from the Hello BPDUs and distribute it to the LANs for which they are designated bridges.

Bridge 1	Bridge 2	Bridge 3
Bridge Priority 32768	Bridge Priority 32768	Bridge Priority 32768
Address	Address	Address
00:00.90:00.00:10	00:00.90:00.00:01	00:00.90:00.00:05
Port 1	Port 1	Port 1
Priority: 128	Priority: 128	Priority: 128
Path Cost: 100	Path Cost: 100	Path Cost: 100
Port 2	Port 2	Port 2
Priority: 128	Priority: 128	Priority: 128
Path Cost: 17857	Path Cost: 17857	Path Cost: 17857
Port 3	Port 3	Port 3
Priority: 128	Priority: 128	Priority: 128
Path Cost: 17857	Path Cost: 17857	Path Cost: 17857

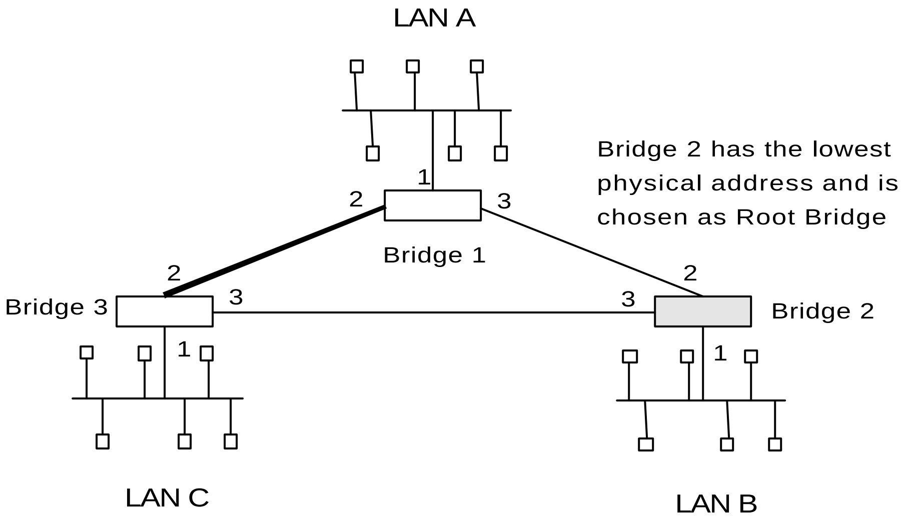


Figure 2-1. Networked LANs Before Spanning Tree

The spanning tree algorithm designates the port connecting Bridge 1 to Bridge 3 (port 2) as a backup port and blocks it from forwarding frames that would cause a loop condition. The spanning tree created by the algorithm using the default values is shown in the Figure 2-2 as the heavy lines connecting Bridge 1 to Bridge 2, and then Bridge 2 to Bridge 3. The root bridge is Bridge 2.

This spanning tree results in poor network performance because the workstations on LAN C can only get to the file server on LAN A indirectly through Bridge 2 rather than using the direct connection between Bridge 1 and Bridge 3.

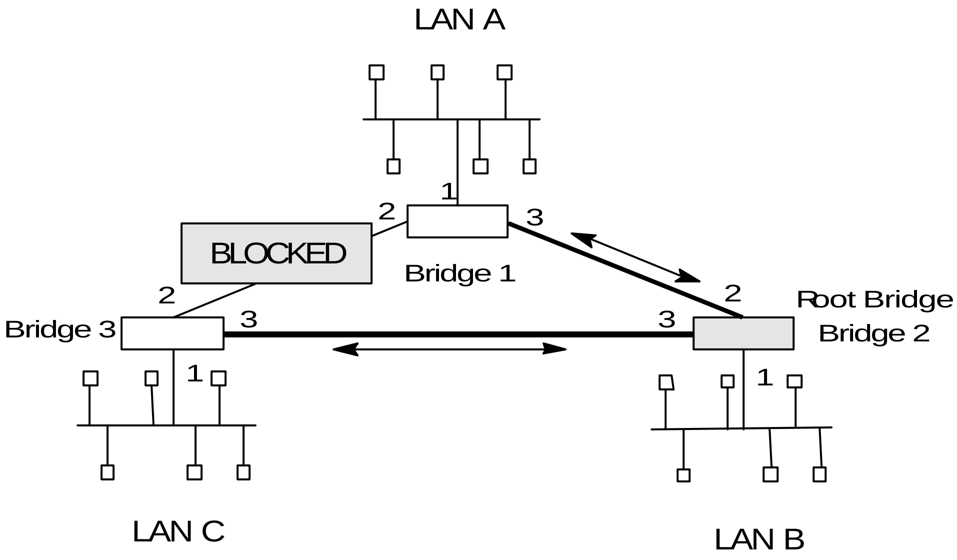


Figure 2-2. Spanning Tree Created with Default Values

Normally this network uses the port between Bridge 2 and Bridge 3 infrequently. Therefore, you can improve network performance by making Bridge 1 the root bridge of the spanning tree. You can do this by configuring Bridge 1 with the highest priority of 1000. The spanning tree that results from this modification is shown in Figure 2-3 as the heavy lines connecting Bridge 1 to Bridge 3 and Bridge 1 to Bridge 2. The root bridge is now Bridge 1. The connection between Bridge 2 and Bridge 3 is now blocked and serves as a backup data path.

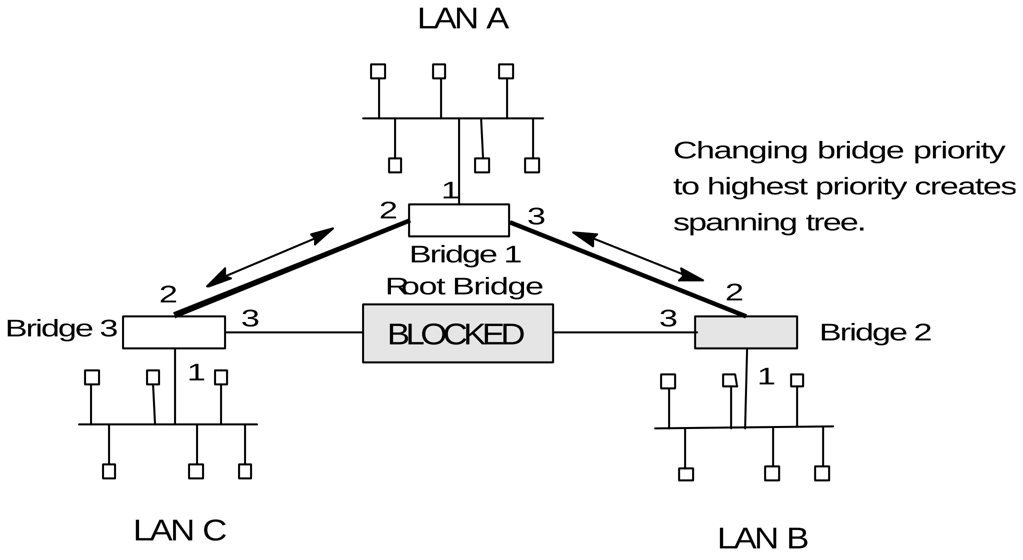


Figure 2-3. User-adjusted Spanning Tree

7. Spanning Tree Bridges and Ethernet Packet Format Translation

The SSTB protocol forwards packets in accordance with IEE Standard 802.1D-1990 Media Access Control (MAC) bridges. It can create a transparent bridge between any combination of Ethernet/ IEEE 802.3 networks, either locally or via serial lines. The protocol also provides appropriate header translation for Ethernet packets.

An Ethernet/IEEE 802.3 network can simultaneously support the Ethernet data link layer based on the value of the length/type field in the MAC header.

The basic approach consists of translating Ethernet packets to IEEE 802.2 Unnumbered Information (UI) packets using the IEEE 802 SNAP SAP. The SNAP Protocol Identifier has the Organizationally Unique Identifier (OUR) of 00-00-00, with the last two bytes being the Ethernet *type* value.

The translation is done when a frame is sent on a LAN. The original frame format is preserved across serial lines.

Chapter 3
Using Source Route Bridging (SRB)

1. About SRB

Source Route Bridging (SRB) is a method of forwarding frames through a bridged network in which the source station identifies the route that the frame will follow. In a distributed routing scheme, routing tables at each bridge determine the path that data takes through the network. By contrast, in a source route bridging scheme, the source station defines the entire route in the transmitted frame.

SRB provides local bridging over 4 and 16 Mbps Token Rings. See Figure 3-1. It can also connect remote LANs through a telecommunications link operating at speeds up to E1.

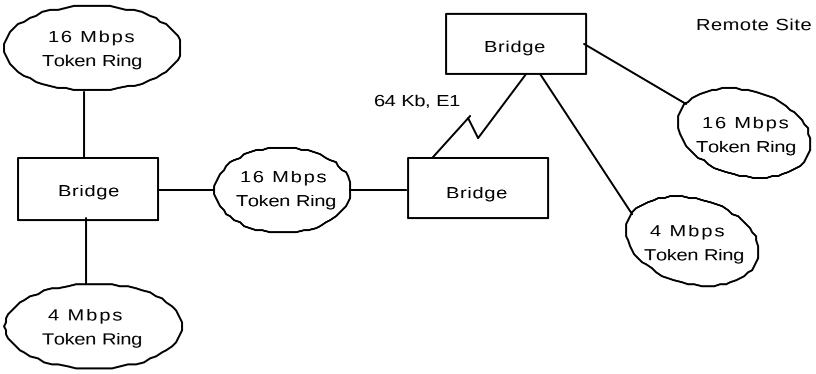


Figure 3-1. Source Routing Bridge Connectivity Sample

Among its features, the source routing bridge provides:

- *IBM compatibility.* The bridge is compatible with the IBM source routing bridge. It can connect IBM PC LANs running systems such as OS/2 and NetBIOS. It can also carry IBM SNA traffic between PC LANs and mainframes.
- *Performance and speed.* Because bridging occurs at the data-link layer instead of the network layer, packet conversion and address table maintenance are not necessary. The means less overhead and higher-speed routing decisions.
- *Bridge tunneling.* By encapsulating source routing packets, the bridge dynamically routes these packets through internetworks to the desired destination end station without degradation or network size restrictions.
- *FCS preservation.* Teldat bridges preserve Frame Check Sequence of the Specifically Routed Frames (SRF). This protects against data corruption of the bridged frames.

Source routing end stations see this path (the *tunnel*) as a single hop, regardless of the network complexity. This helps overcome the usual seven-hop distance limit encountered in source routing configurations. This feature also lets you connect source routing end stations across non-source routing media (e.g. Ethernet networks).

2. Enabling SRB

The following information outlines the initial steps required to enable the SRB bridging option offered by the ASRT bridge.

- **Bridge.** Enables bridging on all LAN interfaces. You can include WAN interfaces (serial lines) by using the **PORT** command.
- **No transparent** *port#*. Disables transparent bridging on all ports.
- **Source-routing** *port# segment#/bridge#*. When source-routing is enabled on more than two ports, an additional segment number is required to assign an internal virtual segment needed for 1:N SRB configurations.

If source routing is the only feature you want, disable transparent bridging on the interfaces.

Do *not* include interfaces that traditionally do not support source routing. For example, if transparent bridging is disabled and source routing is enabled on an Ethernet port, the bridging facility is disabled for this port.

After completing the procedures just described, you can enter **LIST BRIDGE** to verify your configuration.

If you want to make changes to the configuration, see **Chapter 7** '**ASRT Configuration**' of this guide. After you finish changing the configuration, you must restart the router for the new configuration to take effect.

3. How SRB Works

As mentioned, the source station defines the entire route in the transmitted frame in a source routing configuration. The source routing bridge is dynamic. Both end stations and bridges participate in the route discovery and forwarding process. The following steps describe this process:

1. A source station sends out a transparent frame and finds that the frame's destination is not on its own (local) segment or ring.
2. The source station builds a *route discovery* broadcast frame and transmits it onto the local segment.
3. All bridges on the local segment capture the route discovery frame and send it over their connected networks.
4. As the route discovery frame continues its search for the destination end station, each bridge that forwards it adds its own bridge number and segment number to the routing information field (RIF) in the frame. As the frame continues to pass through the bridge network, the RIF compiles a list of bridge and segment number pairs describing the path to the destination. When the broadcast frame finally reaches its destination, it contains the exact sequence of addresses from source to destination.
5. When the destination end station receives the frame, it generates a response frame including the route path for communication. Frames that wander to other parts of the bridged network (accumulating irrelevant routing information in the meantime) never reach the destination end station and no station ever receives them.
6. The originating station receives the learned-route path. It can then transmit information across this established path.

4. SRB Frame Formats

As mentioned bridges interconnect LANs by relaying data frames, specifically MAC frames between the separate MAC entities of the bridged LANs. MAC frames provide the necessary forwarding information in the form of source and destination addresses. This information is essential for the successful transmission and reception of data.

In source routing, the data-frame-forwarding decision is based on routing information within the frame. Before forwarding the frame, end stations have obtained the route to the destination station by *route discovery*. The source station that originates the frame designates the route that the frame will travel by embedding a description of the route in the RIF of the transmitted frame. A closer look at the various types of source routing bridge frames will help to explain further how the bridge obtains and transmits this routing information.

Since source routing MAC frames contain routing information necessary for data communication over multi-ring environments, they differ slightly in the format for the typical Token Ring MAC frames. The presence of a 1 in the RII within the source address field indicates that an RIF containing routing information follows the source address. Figure 3-2 provides a closer look at the format of the source address field of a source routing frame.

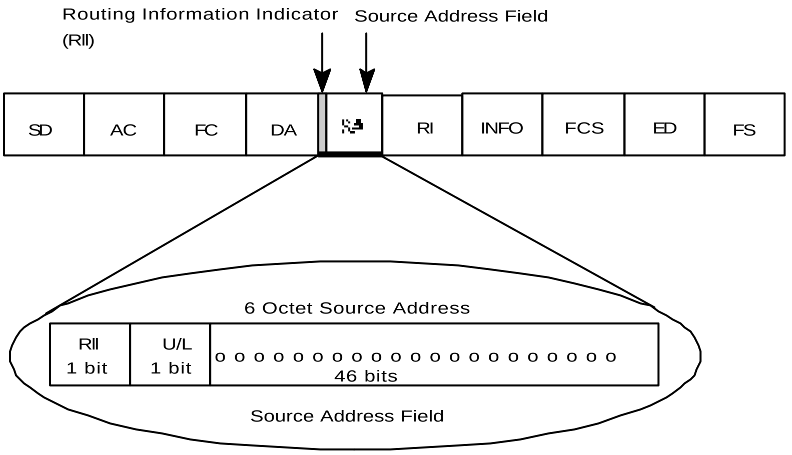


Figure 3-2. 802.5 Source Address Format

When the RII in the source address field is set to 1, an RIF is present after the source address. The RIF is required because it provides route information during source routing. It consists of a 2-octet routing control (RC) field and a series of 2-octet route designator (RD) fields. Figure 3-3 provides a closer look at the format of the Routing Information Field.

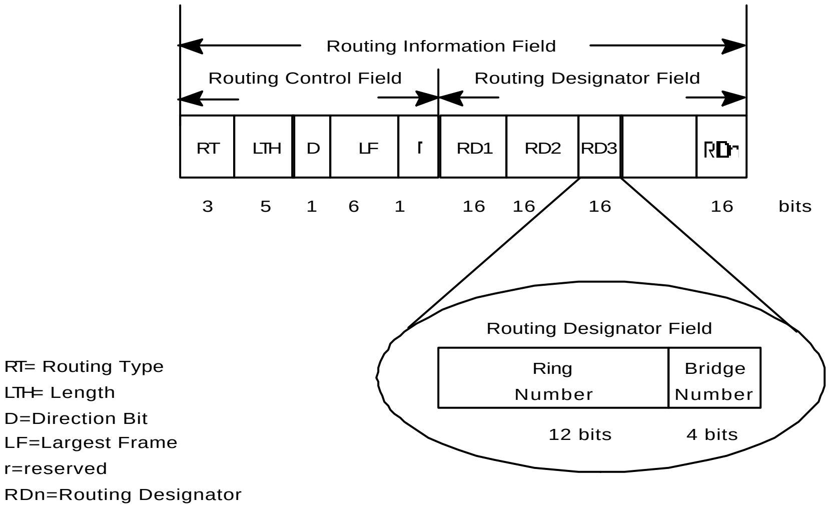


Figure 3-3. 802.5 Routing Information Field

The following information describes each specific field found in the RIF:

- *Routing Type (RT).* Indicates by bit settings if the frame is to be forwarded through the network along a specific route or along a route (or routes) that reaches all interconnected LANs.

Depending on the bit settings in this field the source routing frame can be identified as one of the following types:

- All-Route Explorer frame, ARE (explorer frame)
- Spanning-Tree Explorer frame, STE (explorer frame)
- Specifically-Routed Frame, SRF (data frame)

All-Route explorer frames exist if the RT bits are set to 10x where x is *a don't care* bit. These frames are generated and routed along every non-repeating route in the network (from source to destination). This results in as many frames arriving at the destination end station as there are different routes from the source end station. This routing type can be used as a response to receiving a route discovery frame sent along the spanning tree to the present originating station for all the routes available. The forwarding bridges add routing designators to the frame.

A *spanning tree explorer frame* exists if the TR bits are set to 11x where x is a *don't care* bit. Only spanning tree bridges relay the frame from one network to another. This means that the frame appears only once on every ring in the network and therefore only once at the destination end station. A station initiating the route discovery process may use this frame type. The bridge adds routing designator fields to the frame. It can also be used for frames sent to stations using a group address.

Specifically-routed frames exist if the first RT bit is set to 0. When this is the case, the Route Designator (RD) fields containing specific destination address. During route discovery phase, this type of frame is used as a response to ARE frame. The user data are always carried in the SRF frame format.

- *Length bits (LTH).* Indicates the length (in octets) of the RI field.
- *Direction bit (D).* Indicates the direction the frame takes to traverse the connected networks. If this bit is set to 0, the frame travels the connected networks in the order in which they are specified in the routing information field (e.g. RD1 to RD2 to …. to RDn). If the direction bit is set to 1, the frame travels the networks in the reverse order.

- *Largest Frame Bits (LF).* Indicates the largest frame size of the INFO field that can be transmitted between two communicating end stations on a specific route. The LF bits are meaningful only for STE and ARE frames. In an SRF, the bridge ignores the LF bits and cannot alter them. A station originating an explorer frame sets the LF bits to the maximum frame size it can handle. Forwarding bridges set the LF bits to the largest value that does not exceed the minimum of:
    - The indicated value to the received LF bits
    - The largest MAC Service Data Unit (MSDU) supported by the port from which the frame was received
    - The largest MSDU supported by the port on which the frame is to be transmitted

    The destination station may further reduce the LF value to indicate its maximum frame capacity.

    LF bit encodings are made up of a 3-bit base encoding and a 3-bit extended encoding (6 bits total). The SRT bridge contains an LF mode interpretation indicator so the bridge can select either base or extended LF bits. When the LF mode interpretations indicator is set to *base mode,* the bridge sets the LF bits in explorer frames with the largest frame *base* values. When the LF mode indicator is set to *extended mode,* the bridge sets the LF bits in explorer frames with the largest frame *extended* values.
- *Route Designator fields (RDn),* indicates the specific route through the network according to the sequence of the RD fields. Each RD field contains a unique network 12-bit ring number and 4-bit bridge number that differentiates between two or more bridges when they connect the same two rings (parallel bridges). The last bridge number in the routing information field has a null value (all zeros).

5. The Spanning Tree Explore Option

The *spanning tree explore* option lets you select a single route to a destination when your network has two or more bridges connecting the same LANs. With this feature enabled, only the bridges you select receive STE frames. Not to be confused with the spanning tree protocol, this option allows you to simulate a spanning tree network.

5.1. Simulating a Spanning Tree Network

SRB bridges participate in IBM's proprietary Spanning Tree Protocol (STP). Participation in STP allows SRB bridges to prune a meshed network topology to a non-looped spanning tree automatically. For a network with parallel SRB bridges, as shown in Figure 3-4, STP algorithm automatically blocks one of the ports of a bridge (in this example Bridge B). This causes STE frames to be forwarded via Bridge A only. You can configure bridges to not participate in STP and manually enable or disable STP on each port of each bridge. Obviously, use of manual configuration is discouraged, but may be required under certain circumstances.

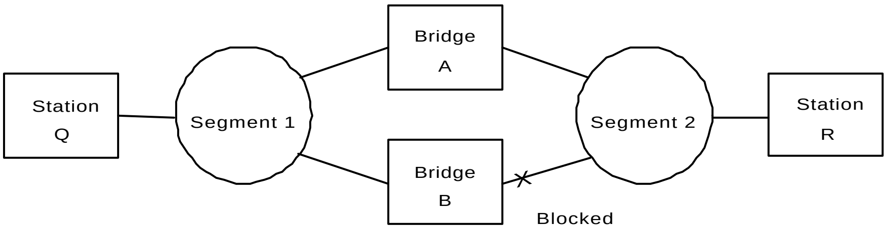


Figure 3-4. Sample Parallel Bridge

6. SRB and Frame Relay

The Frame Relay interface forwards source-routed frames to and from the bridging forwarder provided source routing bridging is enabled on the Permanent Virtual Circuit (PVC).

A destination ring number is configured for each PVC. Some PVC's that are not part of the active data path are blocked in order to maintain the loop-free topology.

Chapter 4
Using Source Route-Transparent Bridge (SR-TB)

1. About SR-TB Conversion

The *Source Route-Transparent Bridge* (SR-TB) conversion option interconnects networks using source route bridging (source route domain) and transparent bridging (transparent abridge domain). It transparently joins both domains. Stations in both domains are not aware of the existence of the SR-TB bridge. Any station on the combined network appears to be in its own domain.

Source routing is available in the SRT model, between adjacent source routing Token Rings. Source-route-only bridges cannot coexist with SRT bridges that link Ethernet and Token Ring LANs. Because a Token Ring end node needs to communicate with an Ethernet node, it must be configured to omit RIFs. But if the end node is configured to omit RIFs, it cannot communicate through ordinary source routing bridges that require that RIF.

SR-TB achieves this functionality by converting frames from the transparent bridging domain to source routing frames before forwarding them to the source routing domain (and vice versa). The bridge does this by maintaining a database of end station addresses, each with its RIF in the source routing domain. It also conducts route discovery on behalf of the end stations present in the transparent bridging domain. It uses route discovery to find the route to the destination station in the source routing domain. It sends frames addressed to an unknown destination in the Spanning Tree Explorer (STE) format.

SR-TB can handle three types of spanning tree:

- A spanning tree formed by a transparent bridge domain
- A spanning tree formed by a source routing bridge domain
- A special spanning tree of all SR-TB bridges

The next sections discuss the operation of SR-TB in more detail.

2. Enabling SR-TB

The information immediately following outlines the initial steps required to enable the SR-TB bridging option offered by the ASRT bridge.

- **Bridge.** Enables bridging.
- **Port.** Enables bridge for a determined interface. Execute this command for all LAN/WAN interfaces over which the bridge is going to operate.
- **No transparent** *port#*. Disables transparent bridging on underlying interfaces.
- **Source-routing** *port#*. Enables source-routing for given ports. When source routing is enabled on more than two ports, an additional segment number is required to assign an internal virtual segment needed for 1:N SRB configurations.
- **sr-tb-conversion** *segment#*. Enables conversion of source-routed frames to transparent frames and vice versa. You must also assign a domain segment number and a domain MTU size to represent the *entire* transparent (Ethernet/FDDI) bridging domain.

After completing the procedures just described, you can enter **LIST BRIDGE** to display the current bridge configuration. This lets you verify and check your configuration.

If you want to make changes to the configuration, see the **Chapter 7 "ASRT Configuration"** of this guide for more details. After you finish making the changes to the configuration, restart the router for the new configuration to take effect.

3. How SR-TB Conversion Works

During SR-TB bridging, a network is partitioned into two or more separate domains. Each domain is made up of a collection of LAN segments interconnected by bridges all operating under a common bridging method. This allows networks composed of two types of domains:

- Source routing
- Transparent bridging

Figure 4-1 shows an example of these domains. With separate domains, each source routing domain has a single-route broadcast topology set up for its bridges. Only bridges belonging to that source routing spanning tree are designated to forward single-route broadcast frames. In this case, frames that carry the single-route broadcast indicator are routed to every segment of the source routing domain. Only one copy of the frame reaches each segment, since the source routing spanning tree does not allow multiple paths between any two stations in the domain.

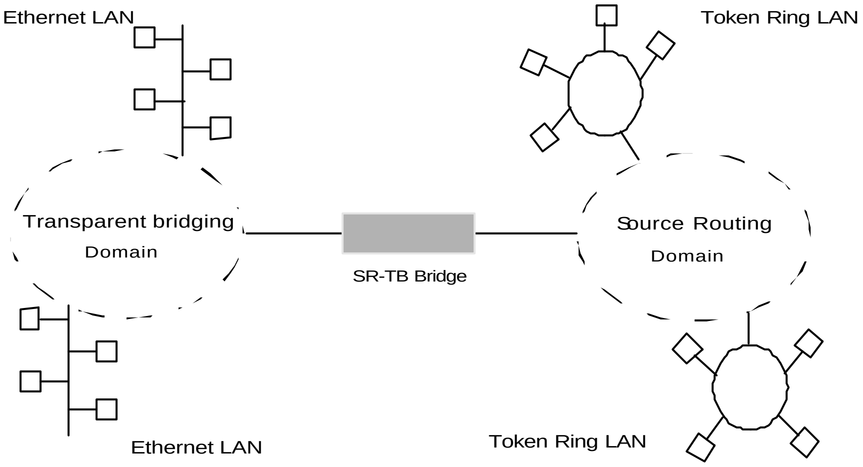


Figure 4-1. SR-TB Bridge Connecting Two Domains

4. Specific Source Routing and Transparent Bridging Operations

SR-TB is a two-port device with a MAC interface assigned to the LAN segment on the source routing side and another assigned to the LAN segment on the transparent bridging side. Each end station reads the appropriate MAC layer for its LAN segment.

On the transparent bridging side, SR-TB operates the same as any other transparent bridge. It keeps a table of addresses for stations it knows are transparent bridging stations. It observes the inter-bridge protocols necessary to create and maintain the network spanning tree since more than one SR-TB joins different domains.

SR-TB forwards a frame received from its transparent bridging station to the source routing side only if it does not find the frame's destination address in the transparent bridging side address table.

On the source routing bridging side, SR-TB combines the functions of a source routing bridge and a source routing end station in a specific way. As a source routing end station, it maintains an association of destination addresses and routing information. It communicates either as an end station for applications in the bridge itself (e.g. network management) or as an intermediary for stations on the transparent bridging side.

SR-TB forwards a frame received from its transparent bridging station to the source routing side of the bridge only if it does not find the frame's destination address in the transparent bridging side address table. Frames transmitted by the bridge's source routing station carry the routing information associated with the bridge, if such information is known and held by the bridge.

As a source routing bridge, SR-TB participates in the route discovery process and in the routing of frames already carrying routing information. The route designator unique to SR-TB consists of the LAN number of the individual LAN on its source routing side and its own individual bridge number.

It also maintains a single LAN number representing all of the LANs on the transparent bridging side. It treats each case of received and forwarded frames differently as described in **Table 4-1**.

Table 4-1 SR-TB Bridge Decision Table

Type of Frame Received	**Action Taken by SR-TB**
Non-routed frame received by the source routing station.	Does not copy or forward frame carrying routing information.
All-routes broadcast frame received by the source routing station.	Copies frame and sets A and C bits of the broadcast indicator in the repeated frame. If destination address is in the transparent bridging table, forwards the frame without routing information on the transparent bridging network. Otherwise, does not forward frame.
Single-route broadcast frame received by the Source Routing station. Bridge *is not* designated as a single-route broadcast bridge.	Does not copy or forward the frame.
Single-route broadcast frame received by the Source Routing station. Bridge *is* designated as single-route broadcast bridge.	Copies frame sets, A and C bits in the broadcast indicator, removes the routing information from the frame, and forwards modified frame to transparent bridging side. Adds its bridge number to saved routing

	information field and the LAN number for transparent bridging side. Changes broadcast indicator to non-broadcast, complements D-bit, and stores this routing information for the source address of the frame.
Non-broadcast frame received by the source routing station	If frame carries specific route, bridge examines the routing information. If SR-TB is part of the route and appears between the LAN number for the source routing side and LAN number for transparent bridge side, copies frame and sets A and C bits in the repeated frame. Forwards frame to the transparent bridging side without routing information. If SR-TB does not already have a permanent route for the source address, saves a copy of the routing information, complements D-bit, and stores saved routing information for the source address of frame.
Frame received from the Transparent bridging side.	To forward frame to the source routing side, first determines if it has routing information associated with the destination address carried in the frame. If yes, adds routing information to the frame, sets the R11 to 1, and queues the frame for transmission on the source routing side. If no, adds a routing control field to the frame containing an indicator for single-route broadcast and two route designators containing the first two LAN numbers and its own individual bridge number.

4.1. SR-TB Bridging: Examples

SR-TB interconnects source routing domains with transparent bridging domains by transparently joining the domains. During operation, stations in both domains are not aware of the existence of SR-TB. From the end station's point of view, any station on the combined network appears to be in its own domain.

The following sections provide specific examples of frame forwarding during SR-TB bridging. These examples assume that SR-TB is designated as a single-route broadcast bridge. Figure 4-2 provides the following information to accompany the situations described in each section:

- D is the bridge's own bridge number
- X is the LAN number for the LAN on the source routing side
- Y is the LAN number for the LAN on the transparent bridging side
- A,B,C, and D are end stations

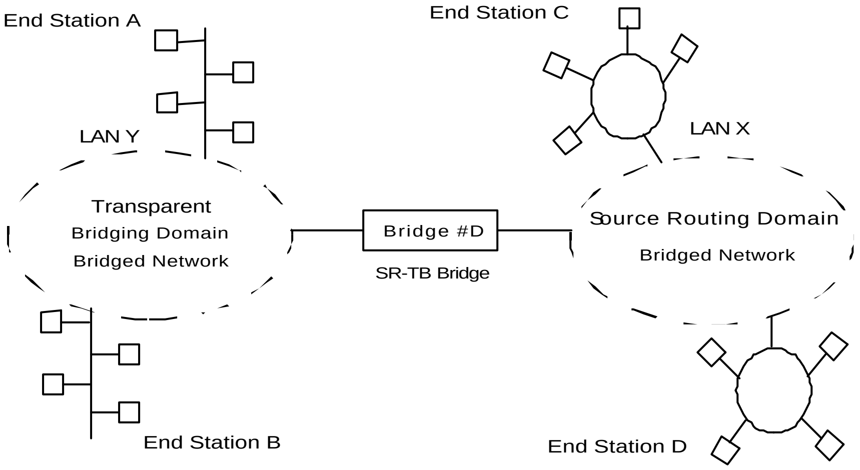


Figure 4-2. SR-TB Bridging Examples

a) Example 1: Frame sent from end station A to end station B

When SR-TB receives a frame with a source address of end station A and destination address of end station B, it puts end station A's address into its transparent bridging side address table. This table contains the addresses of stations known to be on the transparent bridging side of the bridge. This is normal behavior for transparent bridging.

If end station B's address is in the transparent bridging side's address table, SR-TB does not forward the frame. If end station B's address is not in the transparent bridging side's address table and not in the source routing side's address table, SR-TB does not know its location. In this case, SR-TB forwards the frame on the source routing side as a single-route broadcast with no request for route-explorer return. Any frame end station B sends (regardless of its destination) causes its address to be added to the transparent bridging address table. This prevents future forwarding of frames addressed to end station B to the source routing side.

b) Example 2: Frame sent from end station A to end station C

In this example, end station A's address is treated the same as in the previous example. Since end station C's address is not in the transparent bridge address table, SR-TB forwards the frame on the source routing side.

The bridge then looks for end station C's address in its source routing address table. This table contains all known addresses and related routing information for stations on the source routing side of the bridge. If C's address is in the source routing table, the bridge forwards the frame using the routing information in the address table. If C's address is not in the source routing table (or if it appears but has null routing information), the bridge forwards the frame on the source routing side as a single-route broadcast with no request for route-explorer return.

When end station C receives this frame, it enters end station A's address in its source routing table together with the reverse direction of the route built from the SR-TB bridge and marks it as a temporary entry. When end station C later tries to send a frame to end station A, it uses this specific route, and because the route is marked as temporary, sends it as a non-broadcast route with a request for route-explorer return.

When the returning frame arrives, SR-TB forwards it on the transparent bridge side without routing information but puts the route to end station C into the source routing table as a temporary route. This further causes the network management entity (SMT) to send a route-explorer frame with an all-routes broadcast setting back to end station C. This lets end station C select the optimal routing for frames addressed to end station A, which SR-TB then puts into its source routing table as a permanent route.

c) Example 3: Frame sent from end station C to end station D

If the frame is sent as a non-broadcast and crosses over the segment to which the SR-TB bridge is attached, the bridge scans the R11 filed for the routing sequence (LAN X to Bridge Q to LAN Y). It cannot find the sequence and so does not forward the frame.

If the frame is sent as a single-route broadcast, the bridge discards the frame if it already knows that the end station D is on the source routing side. If it does not know that, it forwards the frame to the transparent bridging side (minus the routing information), and adds Q to Y to the routing information. Finally, it saves the routing information for end station C as a temporary route in the source routing table with a non-broadcast indicator and the direction bit complemented.

If the frame is sent as an all-routes broadcast, SR-TB discards the frame (because end station D's address is not present in the transparent bridging address table) and makes sure that end station C's address is in the source routing table.

d) Example 4: Frame sent from end station C to end station A

If the frame is sent non-broadcast, SR-TB scans the R11 field for the routing sequence (X to Q to Y). When it finds it, it forwards the frame to the transparent bridging side. It also stores the routing information for end station C.

If the frame is sent as a single-route broadcast, SR-TB forwards it (minus the routing information) to the transparent bridging side and adds Q to Y to the routing information. It also sets the non-broadcast indicator, complements the direction bit, and enters the routing information for C's address in its source routing table. If a temporary entry for end station C already exists in the source routing table, SR-TB updates the routing information.

If the frame is sent as an all-routes broadcast, SR-TB discards it, but makes sure that end station C's address is in the source routing table.

5. SR-TB and Frame Relay

The Frame Relay interface supports SR-TB bridging by forwarding all bridged frames to the appropriate bridging forwarder, provided bridging is enabled on the Permanent Virtual Circuit (PVC).

Chapter 5
Miscellaneous Bridge Features

1. Protocol Filtering

A single platform can perform both bridging and routing. Protocol Filtering determines whether the incoming data is routed or bridged, based on the contents of the destination address field of incoming frames.

Table 5-1 shows how the destination address field determines “Bridge or Route?” question.

If destination MAC Addressing Contains:	**Action the Bridge Takes**
Interface Address	Passes the frame to the configured protocol that routes the frame.
Multicast or Broadcast Address	If the frame belongs to a configured protocol, passes the frame to the protocol forwarder. Otherwise, bridges the frame.
Other Unicast	If the frame belongs to a configured protocol, discards the frame. Otherwise, bridges the frame.

Table 5-1. Route/Bridge Decision Table

2. IBM RT Feature for SNA Traffic

Some IBM PCs (RT PC running OS/2/EE) run SNA over Ethernet Type 2 instead of 802.3 Ethernet. This requires an additional header that contains the length of the MAC user data followed by the 802.2 (LLC) header.

You can enable or disable the processing of these frames on a per port basis. If enabled, the bridge learns the source station's behavior and generates the correct frame format. But if there is no information about the station's behavior (multicast or unknown stations). The bridge produces duplicate frames, one in 802.3 and 802.2 format, and the other with the IBM-RT header.

3. UB Encapsulation of XNS Frames

XNS Ethernet frames use Ethertype 0x0600. When translated to Token Ring format, these frames get SNAP as specified in IEEE 802.1H. Because some Token Ring end stations use the Ungermann-Bass OUI in the SNAP for such frames, there is a configuration switch to activate this encapsulation.

4. Multiple Spanning Tree Protocol Problems

ASRT lets you extend spanning tree protocol options to cover as many configuration options as possible. The next sections describe these features.

4.1. Multiple Spanning Tree Protocol Problems

Bridging technology employs different spanning tree algorithms to support different bridging methods. The common purpose of each algorithm is to produce a loop-free topology.

In the spanning tree algorithm used by Transparent Bridges (TB), Hello Bridge Protocol Data Units (BPDUs) and Topology Change Notification (TCN) BPDUs are sent in a transparent frame to well-known group addresses of all participating media (Token Ring, Ethernet, FDDI, etc.). Tables are built from this exchanged information and a loop free topology is calculated.

SRB uses transparent frames to determine a loop free topology. The algorithm sends Hello BPDUs in a transparent frame to a well-known functional address. SRB bridges do not use TCN BDPUs. The port state setting created as a result of this spanning tree algorithm does not affect All Route Explorer (ARE) Frame and Specifically Routed Frame (SRF) traffic.

In bridging configuration using IBM 8209 bridges, a different spanning tree method is used to detect parallel 8209 bridges. This algorithm uses Hello BPDUs sent as STE frames to IEEE 802.1d group address on the Token Ring. On the Ethernet, Hello BPDUs are sent as transparent frames to the same group address. This method allows 8209s to build spanning trees with transparent bridges and other IBM 8209 bridges. It does not participate in the SRB spanning tree protocol however, and Hello BPDUs sent by SRBs are filtered. As such, there is no way to prevent the 8209 from becoming the root bridge. If the 8209 bridge is selected as the root, then traffic between two STB domains may have to pass through Token Ring/SRB domains.

4.2. Enhanced STP

The enhanced STP bridging feature allows you to further extend the Spanning Tree protocol. Based on the bridge personality, it allows bridges to participate in the appropriate STP. Previously, SRB bridges allowed only manual configuration of a loop-free tree over the Token Ring. This was the only mechanism to prevent loops in the case of parallel SRB bridges. With the addition of the enhanced STP feature the following spanning tree algorithm combinations are possible:

- Pure Transparent Bridge (STB) - IEEE 802.1D Spanning Tree protocol.
- Pure Source Route Bridge (SRB) - IBM SRB Spanning Tree protocol.
- Transparent and Source Route Bridges as separate entities - IEEE 802.1d Spanning Tree protocol for STB and manual configuration for SRB loop-free topology.
- SR-TB Bridge - IEEE 802.1D Spanning Tree protocol for STB ports and IBM 8209 BPDUs on SRB ports to form a single tree of STBs and SR-TBs. SRB Hello BPDUs are allowed to pass on the DR domain but are not processed.

IBM 8209 bridges filter such frames but this is allowed as it is a two-port bridge with the other port being a transparent bridge port.

- ASRT Bridge - IEEE 802.1D Spanning Tree protocol is used to make a tree with STBs and SRT bridges. 8209-like BPDUs are also generated on all SRB interfaces to make tree with SR-TB and IBM 8209 bridges.

These Hello BPDUs are processed as soon as they are received. This causes two Hello BPDUs to be generated and received on all SR and STB interfaces. Since both Hello BPDUs carry the same information, there is no conflict of port information. This lets the ASRT bridge create a spanning tree with IBM 8209 and SR-TB bridges along with other STBs bridges.

Chapter 6
Using IP Tunneling

1. Bridging IP Tunnel

Bridging IP tunnel is another feature of the ASRT bridging software. With the bridging tunnel feature enabled, the software encapsulates packets in the TCP/IP packets. To the router, the packet looks like a TCP/IP packet. Once a frame is encapsulated in an IP envelope, the IP forwarder is responsible for selecting the appropriate network interface based on the destination IP address. This packet can be routed dynamically through large internetworks without degradation or network size restrictions.

The IP tunnel appears to the bridge as one of the bridge ports using IP as a means of input/output device. On the tunnel bridge port you can configure STB, or SRB bridge behavior.

In SRB configuration, IP tunnel helps overcome the usual 7-hop distance limit encountered in source routing configurations. It also lets you connect source-routing end stations across non-source-routing media, such as Ethernet networks.

The bridging tunnel also reduces the large amounts of overhead that source routing causes in wide area networks (WANs).

Finally, it reduces source-routing's sensitivity to WAN faults and failures (if a path fails, all systems must restart their transmissions).

End stations see this path or tunnel, as a single hop, regardless of the complexity of the internetwork. Figure 6.1 shows an example of an IP internetwork using the tunnel feature in its configuration.

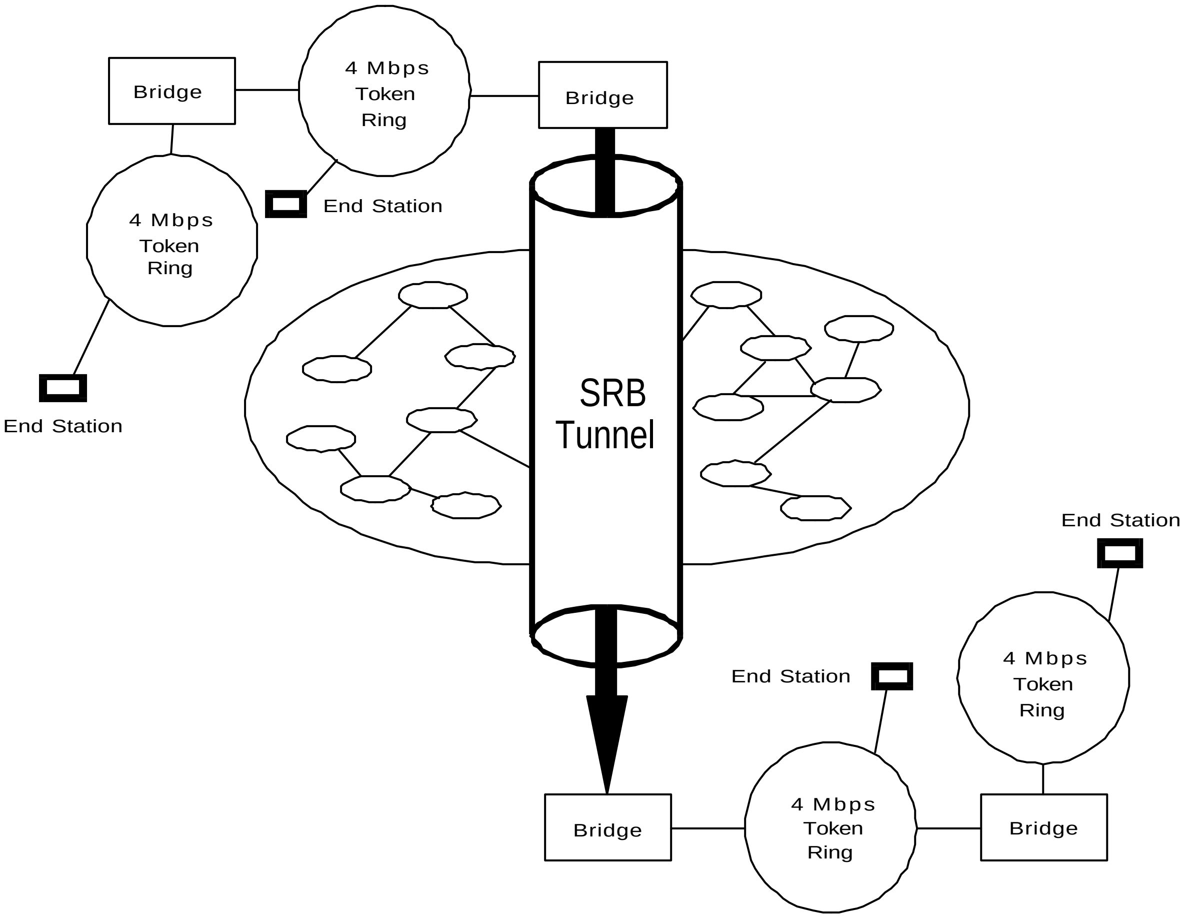


Figure 6.1. End Stations See Routing Across Complex IP Internet as One Hop

The bridges participating in tunneling treat the IP Internet as one of the bridge segment. When the packet reaches the destination interface, the TCP/IP headers are automatically removed and the inner packet proceeds as a standard source-routing packet.

1.1. Encapsulation and OSPF

A major benefit of the encapsulation feature is the addition of the OSPF dynamic routing protocol to the routing process. OSPF offers the following benefits when used with encapsulation:

- *Least-cost Routing.* OSPF accesses the fastest path (tunnel) with the fewest delays, allowing network administrators to distribute traffic over the least expensive route.
- *Dynamic Routing.* OSPF looks for the least-cost path, detects failures, and reroutes traffic with low overhead.

With OSPF, tunnels automatically manage paths inside the internetwork. If a line or bridge fails along the path then the tunnel bridge automatically reroutes traffic along a new path. If a path is restored, the tunnel automatically updates to the best path. This rerouting is completely transparent to the end stations.

Chapter 7
ASRT Configuration

1. ASRT Configuration View

This section describes the ASRT configuration commands. They allow you to specify parameters for the ASRT bridge and its interfaces. They also allow you to enable, and configure the NetBIOS.

To display the `ASRT config>` prompt:

```
Config>PROTOCOL ASRT

-- ASRT Bridge user configuration --
ASRT config>
```

To access the NetBIOS configuration commands, enter **NETBIOS** at the `ASRT config>` prompt to get the `NetBIOS config>` prompt.

```
ASRT config>NETBIOS

NetBIOS Support User Configuration

NetBIOS config>
```

2. ASRT Configuration commands

2.1. ? (HELP)

List the commands available from the current prompt. After a specific command, lists its options.

Syntax:

```
ASRT config>?
```

Example:

```
ASRT config>?
ADDRESS                     Add unique station address entries
BAN                         Displays the BAN config> prompt
BRIDGE                      Bridging functionality
CHANGE                      Changes source routing bridge and segment numbers
DLS                         DLSw over the bridge
DUPLICATE                   Creation of duplicate frames in mixed environments
ETHERTYPE-IBMRT-PC          Translation of SNA frames to Ethernet 2 format
FA-GA-MAPPING               Group address to functional address (and vice versa)
IBM8209-SPANNING-TREE       Participate in spanning tree protocols with IBM 8209
LIST                        Displays bridge configurations
MAPPING                     Functional address to group address mapping
NAME-CACHING                Enter the Name Caching facility configuration menu
NETBIOS                     Displays the Netbios config> prompt
NO
PORT                        Adds a LAN/WAN port to the bridging configuration
PROTOCOL-FILTER             Filter packets based on their protocol type
SET                         Configures several bridge parameters
SOURCE-ROUTING              Source routing on a given port
SPANNING-TREE-EXPLORER      Port propagates spanning tree explorer frames
SR-TB-CONVERSION            Source-routing frame to transparent and vice versa
STP                         STP participation
TRANSPARENT                 Transparent bridging functionality on the given port
TREE                        STP participation for the bridge on a per-port basis
UB-ENCAPSULATION            Ungermann-Bass OUI encapsulation for XNS frames
EXIT
ASRT config>
```

2.2. ADDRESS addr-value

Adds unique station address entries to the permanent filtering database.

Permanent database entries are not destroyed by the power off/on process and are immune to the aging settings. Dynamic entries cannot replace permanent entries.

The *addr-value* is the MAC address of the desired entry. It can be an individual, multicast, or broadcast address. You can also specify the outgoing forwarding port map for each incoming port.

Syntax:

```
ASRT config>ADDRESS <mac-address>
 default              Create a new address

 source-add-filt      Source Address Filtering Applies

 no
        source-add-filt      Source Address Filtering Applies

 bridge               bridge address configuration
        all-same-port         Use all output port mapping for all input Ports
        same-mapping          Use same output port mapping for all input  Ports
        diferent-mapping      Output port mapping for one input port
```

default Sets destination address filtering for that entry. Yes causes filtering of any frames that contain this address as a destination address, no matter which port it came from.

all-same-port Yes creates *one* outgoing port map for all incoming ports rather than allowing for mapping only to specific ports. No causes further prompting.

same-mapping Yes creates an outgoing port map that includes all ports. Thus, when a frame with this address is received, it is forwarded to all outgoing forwarding ports except for the incoming port. The following are examples of how this is done according to the port map:

If a frame is received on *port 1* and the port map indicates 1 (for port 1), the frame is filtered.

If the same frame is received on *port 2* and the port map indicates 1 (for port 1), the frame is forwarded to port 1.

If a frame is received on port 1 and the matching address entry's port map indicates 1, 2, or 3, the frame is forwarded to ports 2 and 3.

If the port map indicates no port (NONE/DAF) then the frame is filtered. This is known as destination address filtering (DAF).

If no address entry is found to match the received frame, it is forwarded to all the forwarding ports (except the source port).

different-mapping. Associates an address entry with that specific bridge port. Yes maps the address to the specified port so that port is included in that address entry's port map. No skips address mapping for that port.

source-add-filt Allows port-specific address filtering. Yes discards frames received with source addresses matching address entries in the filtering database with source address filtering enabled. This lets a network manager isolate an end station by not allowing traffic to be bridged.

The following sections present examples of how to use **ADDRESS** to manage address entries.

Enabling destination address filtering for entry

```
ASRT config>ADDRESS 000000334455 default
ASRT config>
```

After adding the address, verify its status by entering **LIST RANGE**. The example below show that no port map exists for that entry (in bold) and that Destination Address Filtering (DAF) is on.

```
ASRT config>LIST RANGE
Start-Index[1]?
Stop-index[18]?
ADDRESS                 ENTRY TYPE      PORT MAP
=======                 ==========      ========
01-80-c2-00-00-00       REGISTERED      Input Port:  ALL PORTS
                                        Output ports:

01-80-c2-00-00-01       RESERVED        NONE/DAF
01-80-c2-00-00-02       RESERVED        NONE/DAF
01-80-c2-00-00-03       RESERVED        NONE/DAF
01-80-c2-00-00-04       RESERVED        NONE/DAF
01-80-c2-00-00-05       RESERVED        NONE/DAF
01-80-c2-00-00-06       RESERVED        NONE/DAF
01-80-c2-00-00-07       RESERVED        NONE/DAF
01-80-c2-00-00-08       RESERVED        NONE/DAF
01-80-c2-00-00-09       RESERVED        NONE/DAF
01-80-c2-00-00-0a       RESERVED        NONE/DAF
01-80-c2-00-00-0b       RESERVED        NONE/DAF
01-80-c2-00-00-0c       RESERVED        NONE/DAF
01-80-c2-00-00-0d       RESERVED        NONE/DAF
01-80-c2-00-00-0e       RESERVED        NONE/DAF
01-80-c2-00-00-0f       RESERVED        NONE/DAF
03-00-00-00-80-00       RESERVED        NONE/DAF
00-00-00-33-44-55       PERMANENT       NONE/DAF
ASRT config>
```

Creating separate output port maps for an address entry that has more than one input port.

```
ASRT config>ADDRESS 000000012345 bridge diferent-mapping 1 1
ASRT config>ADDRESS 000000012345 bridge diferent-mapping 1 2
ASRT config>ADDRESS 000000012345 bridge diferent-mapping 2 1
ASRT config>ADDRESS 000000012345 bridge diferent-mapping 2 2
ASRT config>ADDRESS 000000012345 source-add-filt
ASRT config>
```

After adding the address, verify its status by entering **LIST RANGE**. The example below shows an entry (in bold) that has ports 1 and 2 as input ports and has separate port maps for both input ports. Source Address Filtering (SAF) is also enabled.

```
ASRT config>LIST RANGE
Start-Index[1]?
Stop-index[18]?
ADDRESS                 ENTRY TYPE      PORT MAP
=======                 ==========      ========
=======                 ==========      ========
                                        Output ports:

01-80-c2-00-00-01       RESERVED        NONE/DAF
01-80-c2-00-00-02       RESERVED        NONE/DAF
01-80-c2-00-00-03       RESERVED        NONE/DAF
01-80-c2-00-00-04       RESERVED        NONE/DAF
```

```
01-80-c2-00-00-05       RESERVED        NONE/DAF
01-80-c2-00-00-06       RESERVED        NONE/DAF
01-80-c2-00-00-07       RESERVED        NONE/DAF
01-80-c2-00-00-08       RESERVED        NONE/DAF
01-80-c2-00-00-09       RESERVED        NONE/DAF
01-80-c2-00-00-0a       RESERVED        NONE/DAF
01-80-c2-00-00-0b       RESERVED        NONE/DAF
01-80-c2-00-00-0c       RESERVED        NONE/DAF
01-80-c2-00-00-0d       RESERVED        NONE/DAF
01-80-c2-00-00-0e       RESERVED        NONE/DAF
01-80-c2-00-00-0f       RESERVED        NONE/DAF
03-00-00-00-80-00       RESERVED        NONE/DAF
00-00-00-01-23-45       PERM/SAF        Input Port:  1
                                        Output ports:  1, 2
                                        Input Port:  2
                                        Output ports:  3, 4

ASRT config>
```

Creating a single output port map for all incoming ports associated with an address entry

```
ASRT config>ADDRESS 000000556677 bridge same-mapping 1
ASRT config>ADDRESS 000000556677 bridge same-mapping 2
ASRT config>ADDRESS 000000556677 bridge same-mapping 4
```

After adding the address, verify its status by entering **LIST RANGE**. The example below shows an entry (in bold) that has a single port map for all incoming ports. Source Address Filtering (SAF) is also enabled.

```
ASRT config>LIST RANGE
Start-Index[1]?
Stop-index[19]?
ADDRESS                 ENTRY TYPE      PORT MAP
=======                 ==========      ========
01-80-c2-00-00-00       REGISTERED      Input Port:  ALL PORTS
                                        Output ports:

01-80-c2-00-00-01       RESERVED        NONE/DAF
01-80-c2-00-00-02       RESERVED        NONE/DAF
01-80-c2-00-00-03       RESERVED        NONE/DAF
01-80-c2-00-00-04       RESERVED        NONE/DAF
01-80-c2-00-00-05       RESERVED        NONE/DAF
01-80-c2-00-00-06       RESERVED        NONE/DAF
01-80-c2-00-00-07       RESERVED        NONE/DAF
01-80-c2-00-00-08       RESERVED        NONE/DAF
01-80-c2-00-00-09       RESERVED        NONE/DAF
01-80-c2-00-00-0a       RESERVED        NONE/DAF
01-80-c2-00-00-0b       RESERVED        NONE/DAF
01-80-c2-00-00-0c       RESERVED        NONE/DAF
01-80-c2-00-00-0d       RESERVED        NONE/DAF
01-80-c2-00-00-0e       RESERVED        NONE/DAF
01-80-c2-00-00-0f       RESERVED        NONE/DAF
03-00-00-00-80-00       RESERVED        NONE/DAF
00-00-00-33-44-55       PERMANENT       NONE/DAF
00-00-00-55-66-77       PERM/SAF        Input Port:  ALL PORTS
                                        Output ports:  1, 2, 4

ASRT config>
```

2.3. BAN

Displays the `BAN config>` prompt. You can access this prompt by entering **BAN** at the `ASRT config>` prompt as shown below.

Syntax:

```
ASRT config>BAN
```

Example:

```
ASRT config>BAN
Boundary Access Node user Configuration
BAN config>
```

2.4. BRIDGE

Enables transparent bridging on all the LAN devices (interfaces) configured in the router. Assigns port numbers to each interface as the previous interface number plus 1. For example, if interface 0 is a LAN device, its port number is 1.

Example:

```
ASRT config>BRIDGE
ASRT config>
```

2.5. CHANGE

Changes source routing bridge and segment numbers in the bridging configuration.

Syntax:

```
ASRT config>CHANGE ?
BRIDGE
SEGMENT
```

a) CHANGE BRIDGE

Changes bridge numbers in the bridging configuration.

Example:

```
ASRT config>CHANGE BRIDGE
Bridge number in hex (1 - 9, A - F)[1]? 2
ASRT config>
```

b) CHANGE SEGMENT

Changes segment numbers in the bridging configuration.

Example:

```
ASRT config>CHANGE SEGMENT
Old segment number in hex(1 - FFF)[1]?
New segment number in hex(1 - FFF)[1]? 2
ASRT config>
```

2.6. DLS

Enables DLSw over the bridge. The router running DLSw looks like a bridge to the end stations.

Example:

```
ASRT config>DLS
ASRT config>
```

2.7. DUPLICATE

Enables the generation of duplicate STE (Spanning Tree Explorer) or TSF (Transparent Spanning Frames) frames. This command is available to offset the **NO DUPLICATE** command. Duplicate frame generation is enabled by default. The **NO DUPLICATE** command may be followed by a frame type of **TSF** or **STE** to specifically enable one of the frame types, or by the frame type **BOTH** which yields the same behavior as not specifying a frame type.

Syntax:

```
ASRT config>DUPLICATE ?
BOTH
STE
TSF
PORT
```

a) DUPLICATE BOTH

Example:

```
ASRT config>DUPLICATE BOTH
Port Number[1]? 2
ASRT config>
```

b) DUPLICATE STE

Example:

```
ASRT config>DUPLICATE STE
Port Number[1]? 2
ASRT config>
```

c) DUPLICATE TSF

Example:

```
ASRT config>DUPLICATE TSF
Port Number[1]? 1
ASRT config>
```

d) DUPLICATE PORT

Example:

```
ASRT config>DUPLICATE PORT
Port Number[1]? 2
ASRT config>
```

2.8. ETHERTYPE-IBMRT-PC

Enables translation of SNA frames to Ethernet 2 format used by IBM RTs running OS/2/EE. See "IBM RT Feature for SNA Traffic" in Chapter 5 for more details.

Example:

```
ASRT config>ETHERTYPE-IBMRT-PC
Port Number[1]? 1
ASRT config>
```

2.9. FA-GA-MAPPING

Enables mapping of group addresses to functional addresses and vice versa. You need this to forward frames between Token Ring and other media (except serial line). In Token Rings, functional addresses are more popular even though they are locally assigned group addresses due to hardware restrictions. Other media commonly use group addresses. Under normal circumstances mapping group addresses to functional address is inevitable. Mapping is enabled by default if you have added mapping addresses. Enable/disable mapping lets you have a choice when it comes to deleting added map records.

Example:

```
ASRT config>FA-GA-MAPPING
ASRT config>
```

2.10. IBM8209_SPANNING_TREE

Allows bridges to participate in spanning tree protocols with IBM 8209 bridges.

Example:

```
ASRT config>IBM8209_SPANNING_TREE
ASRT config>
```

2.11. LIST

Displays information about the complete bridge configuration or about selected configuration parameters.

Syntax:

```
ASRT config>LIST ?
ADDRESS
BRIDGE
FILTERING
MAPPING
PERMANENT
PORT
PROT-FILTER
PROTOCOL
RANGE
```

a) *LIST ADDRESS*

Reads an address entry from the permanent database.

Example:

```
ASRT config>LIST ADDRESS
Address (in 12-digit hex)[]? 000000123456
00-00-00-12-34-56         PERMANENT     Input Port:  ALL PORTS
                                        Output ports:  1, 2
ASRT config>
```

Example:

```
ASRT config>LIST ADDRESS
Address (in 12-digit hex)[]? 001122334455
00-11-22-33-44-55         PERM/SAF      Input Port:  1
                                        Output ports:  1, 2

ASRT config>
```

Address Address entry in 12-digit hexadecimal format.

Entry Type
Permanent The entry is permanent and survives power on/offs or system resets.
Reserved The entry is reserved by the IEEE802.1d committee for future use. Frames to reserved addresses are discarded.
Registered The entry is meant for the bridge itself.
SAF Appears after the entry type if you configure source address filtering.

Input Port The numbers of input port(s) associated with that address entry.

Output Port The numbers of output port(s) associated with that address entry. NONE/DAF indicates that destination address filtering applies because no ports have been selected to be associated with that address entry.

b) LIST BRIDGE

Lists all general information regarding the bridge.

Example:

```
ASRT config>LIST BRIDGE

                 Source Routing Transparent Bridge Configuration
              ====================================================

Bridge:    Enabled                              Bridge behavior: ADAPTIVE SRT
                   +-----------------------------------------+
-------------------|       SOURCE ROUTING INFORMATION        |----------------
                   +-----------------------------------------+
Bridge Number:          01                       Segments:          1
Max ARE Hop Cnt:        14                       Max STE Hop cnt:  14
1:N SRB:                Active                   Internal Segment:  0x001
LF-bit interpret:       Extended
                   +-----------------------------------------+
-------------------|             SR-TB INFORMATION           |----------------
                   +-----------------------------------------+
SR-TB Conversion:        Enabled
TB-Virtual Segment:      0x001                   MTU of TB-Domain:  1350
                   +-----------------------------------------+
-------------------|    SPANNING TREE PROTOCOL INFORMATION   |----------------
                   +-----------------------------------------+
Bridge Address:         Default                Bridge Priority:   32768/0x8000
STP Participation:       IEEE802.1d and IBM-8209
                   +-----------------------------------------+
-------------------|        TRANSLATION INFORMATION          |----------------
                   +-----------------------------------------+
FA<=>GA Conversion:      Enabled                 UB-Encapsulation:  Enabled
DLS for the bridge:      Enabled
                   +----------------------------------------+
-------------------|           PORT INFORMATION             |-----------------
                   +----------------------------------------+
Number of ports added: 2
Port:   1      Interface:    ethernet0/0 Behavior:   STB & SRB   STP: Enabled

Port:   2      Interface:          tnip1 Behavior:    STB Only   STP: Enabled
 Circuit name: test


ASRT config>
```

Bridge Indicates whether the bridge is enabled or disabled.

Bridge Behavior Method of bridging being used. Values are STB for transparent, SRB for source routing, and SR-TB for source routing-transparent conversion bridging.

Bridge Address Bridge address specified by the user (if set).

Bridge Priority A high-order 2-octet bridge address found in the bridge identifier - either the MAC obtained from the lowest number port of the address set by the **SET BRIDGE** command.

Bridge Number Distinguishes between multiple bridges connecting the same two rings.

Number of Source Routing Segments The number of Source Routing bridge segments configured for the Source Routing domain.

SRB: Max ARE/STE Hop cnt The maximum hop count for frames transmitting from the bridge for a given interface associated with source routing bridging.

SR-TB Conversion Indicates whether source routing/transparent bridge frame conversion is enabled or disabled.

TB-Virtual Segment The segment number of the transparent bridging domain.

MTU for TB-Domain The maximum frame size (maximum transmission units) the transparent bridge can transmit and receive.

1:N Source Routing The current state of 1:N Source Routing ACTIVE or NOT ACTIVE.

Internal Virtual Segment Displays the virtual segment number configured for 1:N SRB bridging.

SRB LF-bit interpretation Indicates the largest frame (LF) bit encoding interpretation mode if source routing is enabled in this bridge (BASIC or EXTENDED).

FA-GA conversion Indicates whether FA-GA conversion is enabled or disabled.

Spanning Tree Protocol Participation The types of spanning tree protocols that the bridge participates in.

Number of ports added The number of bridge ports added to the bridging configuration.

Port Number A user-defined number assigned to an interface using the **PORT** command.

Interface Identifies devices connected to a network segment through the bridge. You must add at least two interfaces to participate in bridging. Use 255 for bridging.

Port Behavior Indicates method of bridging being used by that port. The values are STB for Transparent, SRB for Source Routing, and SR-TB for Source Routing-Transparent conversion bridging.

c) LIST FILTERING

Displays the parameters associated to the bridge filter.

Example:

```
ASRT config>LIST FILTERING
Filtering Database Size : 2048
Ageing Time (in seconds): 300
Resolution  (in seconds): 5
ASRT config>
```

Filtering Database Size: number of entries that the bridge filtering database can have.

Ageing Time: time after which the dynamic entries in the filtering database disappear.

Resolution: temporary resolution.

Further information available in the **SET AGE**, and **SET FILTERING** commands.

d) LIST MAPPING

Lists specific address mapping for given protocol.

Syntax:

```
ASRT config>LIST MAPPING ?
DSAP
ETHER
SNAP
```

· LIST MAPPING DSAP

Example:

```
ASRT config>LIST MAPPING DSAP

PROTOCOL TYPE           GROUP ADDRESS           FUNCTIONAL ADDRESS
=============           =============           ==================
aa                      01-02-03-04-05-06        0a:0b:0c:0d:0e:0f

ASRT config>
```

· LIST MAPPING ETHER

Example:

```
ASRT config>LIST MAPPING ETHER

PROTOCOL TYPE           GROUP ADDRESS           FUNCTIONAL ADDRESS
=============           =============           ==================
ffee                    01-01-01-02-02-02        aa:bb:cc:dd:ee:ff

ASRT config>
```

· LIST MAPPING SNAP

Example:

```
ASRT config>LIST MAPPING SNAP

PROTOCOL TYPE           GROUP ADDRESS           FUNCTIONAL ADDRESS
=============           =============           ==================
000000-0800             ab-00-00-02-00-00        c0:00:20:00:00:00

ASRT config>
```

e) LIST PERMANENT

Displays the number of entries in the bridge's permanent database.

Example:

```
ASRT config> LIST PERMANENT
Number of entries in Permanent Database: 19
ASRT config>
```

f) LIST PORT

Displays port information related to ports already configured. The router asks for the Port it wishes to list. If the number is not specified ([-1]) all ports are displayed.

Example:

```
ASRT config>LIST PORT
Port Number[-1]?
Port Id (dec)    : 128: 1, (hex): 80-01
Port State       : Enabled
STP Participation: Enabled
Port Supports    : Transparent Bridging and Source Routing
SRB: Segment Number: 0x002       MTU:  4399     STE Forwarding: Disabled
Duplicates Frames Allowed:   STE: No   , TSF: Yes
Assoc Interface  : ethernet0/0
Path Cost        : 0
IBM RT-PC Ethertype (0x80D5) processing is enabled
-----------------------------------------------------------------------------
Port Id (dec)    : 128: 2, (hex): 80-02
Port State       : Enabled
STP Participation: Enabled
Port Supports    : Transparent Bridging Only
Assoc Interface  : serial0/0  Circuit name: prueba
Path Cost        : 0
-----------------------------------------------------------------------------
ASRT config>
```

Port ID
The ID consists of two parts: the port priority and the port number. In the example, 128 is the priority and 1, 2, and 3 is the port number. In hexadecimal format, the low-order byte denotes the port number and the high order byte denotes priority.

Port State
Whether or not the port is enabled or disabled.

Port Supports
Displays bridging method supported by that port (for example, transparent bridging, source routing bridging).

SRB
Displayed only when SRB is enabled and lists source-routing bridging information. This includes the SRB segment number (in hex), the Maximum Transmission Unit size, and whether the transmission of Spanning Tree Explorer Frames is enabled or disabled.

Duplicate Frames Allowed
Displays a breakdown and count of the types of duplicate frames allowed.

Assoc Interface
Interface name associated with the displayed port, and FR circuit name if applicable.

Path Cost
Cost associated with the port used for possible root path cost. The range is 1 to 65535.

Note: If IBM RT-PC Ethertype processing is enabled, they appear on this display. If it is not enabled, their status does not appear.

g) LIST PROT-FILTER

Reads a current list of the filter protocol types. You can list filters selectively by port or display all ports at once. Port Number selects the bridge port that you want to list.

Example:

```
ASRT config>LIST PROT-FILTER
Port Number[-1]?
No DSAP Filter Records Associated
Protocol Class: ETHER
Protocol Type : 0800
Protocol State: FILTERED
```

```
Port Map      : 1, 2
==========================
No SNAP Filter Records Associated
ASRT config>
```

Port Number Displayed for each port if you list all ports.

Protocol Class Displays protocol class (SNAP, Ether, or DSAP).

Protocol Type Protocol ID in hexadecimal format.

Protocol State Denotes that protocol is being filtered for selected port.

h) LIST PROTOCOL

Displays bridge information related to the spanning tree protocol.

Example:

```
ASRT config>LIST PROTOCOL
Bridge Identifier              : 32768/000000000000 (using port address)
Bridge-Max-Age (in seconds)    : 20
Bridge-Hello-Time (in seconds)   : 2
Bridge-Forward-Delay (in seconds): 15
ASRT config>
```

> ***Note: Each of these bridge-related parameters is also described in detail in previous chapters.***

Bridge Identifiers 8-byte value in ASCII format. If you do not set the bridge address prior to displaying this information, the low order six bytes displayed as zero indicates the default MAC address of a port. When a bridge is selected as the root bridge, it transmits the bridge max age and bridge hello time to all the bridges in the network via the Hello BPDUS.

Bridge-Max-Age Maximum age (period of time) that should be used to time out spanning-tree-protocol-related information.

Bridge-Hello-Time Time interval between Hello BPDUs.

Bridge-Forward-Delay Time interval used before changing to another state (should this bridge become the root).

i) LIST RANGE

Reads a range of address entries from the permanent database. To do this, first determine the size of the database by using the **LIST PERMANENT** command. From this value you can then determine a start index value for your entry range. The start index is one to the size of the database. You can then choose a stop index to display a limited number of entries. This input is optional. If the stop index is not provided the default is the size of the database. Address entries contain the following information:

Example:

```
ASRT config>LIST RANGE
Start-Index[1]? 17
Stop-index[19]? 19
ADDRESS                ENTRY TYPE      PORT MAP
=======                ==========      ========
03-00-00-00-80-00      RESERVED        NONE/DAF
00-00-00-12-34-56      PERMANENT       Input Port:  ALL PORTS
                                       Output ports:  1, 2

00-11-22-33-44-55      PERM/SAF        Input Port:  1
                                       Output ports:  1, 2

ASRT config>
```

Address 6-byte MAC address of the entry.

Entry Type Specifies one of the following types:

- *Reserved* Reserved by the IEEE802.1d committee
- *Registered* Unicast addresses belonging to proprietary communications hardware attached to the box or multicast addresses enabled by protocol forwarders
- *Permanent* Entries entered in the configuration process that survive power on/offs or system resets
- *Static* Entries entered in the monitoring process that do not survive power on/offs or system resets and are ageless
- *Dynamic* Entries learned by the bridge dynamically that do not survive power on/offs or system resets and have an age associated with them
- *Free* Locations in database free to be filled by address entries

Port Map Outgoing port map for all incoming ports.

2.12. MAPPING

Adds a specific functional address to group address mapping for a protocol identifier. Converts address mapping only on destination addresses crossing Token Ring to Ethernet or vice versa.

> ***Note: For every Ethertype mapped value, add the corresponding SNAP-type value. This is necessary for bidirectional mapping.***

dlh-type (Data-link-header type); Options are Destination Service Access Point (DSAP), Ethertype, or Subnetwork Access Protocol (SNAP).

type-field Protocol type field.

Enter the DSAP protocol type in a range of 1 to FE (hexadecimal).

Enter Ethernet (Ether) protocol type in a range of 5DD to FFFF (hexadecimal).

Enter SNAP protocol type in 10-digit hexadecimal format.

ga-address 6-byte (12-digit hexadecimal) group/multicast address.

fa-address Enter functional address in non-canonical format. Functional addresses are locally administered group addresses, most commonly used in Token Ring networks.

The most commonly used values for DECnet group address-to-functional address mapping are the following:

Ethertype	Group Address	Functional Address
6002	ab-00-00-02-00-00	C0:00:20:00:00:00
6003	ab-00-00-03-00-00	C0:00:10:00:00:00
6003	ab-00-00-00-04-00	C0:00:08:00:00:00

SNAP	Group Address	Functional Address
00-00-00-6002	ab-00-00-02-00-00	C0:00:20:00:00:00
00-00-00-6003	ab-00-00-03-00-00	C0:00:10:00:00:00
00-00-00-6003	ab-00-00-00-04-00	C0:00:08:00:00:00

Example 1:

```
ASRT config>MAPPING DSAP
Protocol Type in hex (1 - FF)[1]?
Group-Address (in 12-digit hex)[]? ab0000020000
Functional-Address (in noncanonical 12-digit hex)[]? c00020000000
ASRT config>
```

Example 2:

```
ASRT config>MAPPING ETHER
Protocol Type in hex (5DD - FFFF)[0800]? 6002
Group-Address (in 12-digit hex)[]? ab0000020000
Functional-Address (in noncanonical 12-digit hex)[]? c00020000000
ASRT config>
```

Example 3:

```
ASRT config>MAPPING SNAP
Address (in 10-digit hex)[0000000800]? 0000006003
Group-Address (in 12-digit hex)[]? ab0000030000
Functional-Address (in noncanonical 12-digit hex)[]? c00010000000
ASRT config>
```

2.13. NAME-CACHING

Use the **NAME-CACHING** command to enter the Name Caching facility configuration menu.

Syntax:

```
ASRT config>NAME-CACHING
Name Cache Config>
```

Commands	Function
? (HELP)	Displays all the configuration Name-caching commands, or lists options for specific commands.
DISABLE	Disables Name-caching facility and duplicate frame filtering.
ENABLE	Enables Name-caching facility and duplicate frame filtering.
LIST	Displays the currently implemented Name-caching configurations.
PORT	Selects a specific interface for configuring purposes.
TIMER	Sets the entry idle timer, the server timer, and the time within which duplicate frames are filtered.
EXIT	Exits the Name-caching configuration prompt.

a) ? HELP

Use the **?** (HELP) command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command to list its options.

Example:

```
Name Cache Config>?
DISABLE
ENABLE
LIST
PORT
TIMER
EXIT
```

b) DISABLE

Disables Name-caching facility and duplicate frame filtering.

Syntax:

```
Name Cache Config>DISABLE ?
ADD-NAME-FILTERING
NAME-CACHING
```

· DISABLE ADD-NAME-FILTERING

Disables duplicate frame filtering. Duplicate frame can be ADD-NAME or ADD-GROUP-NAME.

Example:

```
Name Cache Config>DISABLE ADD-NAME-FILTERING
Name Cache Config>
```

· DISABLE NAME-CACHING

Disables Name-caching facility.

Example:

```
Name Cache Config>DISABLE NAME-CACHING
Name Cache Config>
```

c) ENABLE

Enables Name-caching facility and duplicate frame filtering.

Syntax:

```
Name Cache Config>ENABLE ?
ADD-NAME-FILTERING
NAME-CACHING
```

· ENABLE ADD-NAME-FILTERING

Enables duplicate frame filtering. Duplicate frame can be ADD-NAME or ADD-GROUP-NAME.

Example:

```
Name Cache Config>ENABLE ADD-NAME-FILTERING
Name Cache Config>
```

· ENABLE NAME-CACHING

Enables Name-caching facility.

Example:

```
Name Cache Config>ENABLE NAME-CACHING
Name Cache Config>
```

d) LIST

Use the **LIST** command to display the current Name caching configuration settings.

Example:

```
Name Cache Config>LIST

  Server name caching:      Enabled
  Server timeout:           3
  Add name frame filtering: Enabled
  Add name frame timeout:   7
  Entry timeout:            900

Name Cache Config>
```

e) PORT

Use the **PORT** command to select the bridge's port to which name caching commands will be applied.

Example:

```
Name Cache Config>PORT
Port[1]? 2
Name Cache Port Config>
```

The following commands are available at the *Name Cache Port Config>* prompt:

Syntax:

```
Name Cache Port Config>?
DISABLE
ENABLE
LIST
EXIT
```

f) TIMER

Use the **TIMER** command to set the entry idle timer, the server timer, and the time within which duplicate frames are filtered.

Syntax:

```
Name Cache Config>TIMER ?
ADD-NAME
ENTRY
SERVER-RESPONSE
```

· *TIMER ADD-NAME*

Sets the time within which duplicate frames are filtered. The default setting is 7 seconds.

Example:

```
Name Cache Config>TIMER ADD-NAME
Time in seconds (1-32) [7]?
Name Cache Config>
```

· *TIMER ENTRY*

Sets the entry idle timer. If a client and server do not reference the entry's name within this time interval set on this timer, the entry is removed. The default setting is 900 seconds.

Example:

```
Name Cache Config>TIMER ENTRY
Time in seconds (10-65535) [900]?
Name Cache Config>
```

· *TIMER SERVER-RESPONSE*

Sets the server timer. If the server does not respond to a Name-Query within the set time, the entry's RIF and MAC information is made invalid. The default setting is 3 seconds.

Example:

```
Name Cache Config>TIMER SERVER-RESPONSE
Time in seconds (1-16) [3]?
Name Cache Config>
```

g) EXIT

Use the **EXIT** command to return to the ASRT prompt.

Example:

```
Name Cache Config>EXIT
ASRT config>
```

2.14. NETBIOS

Displays the NetBIOS configuration prompt. Enter **NETBIOS** at the `ASRT config>` prompt to display the NetBIOS configuration prompt.

See Chapter 10 "NetBIOS Filtering and Caching Commands", for an explanation for the NetBIOS commands.

Syntax:

```
ASRT config>NETBIOS
```

Example:

```
ASRT config>NETBIOS

NetBIOS Support User Configuration

NetBIOS config>
```

> ***Note: If you have not purchased the NetBIOS feature, you receive the following message if you use this command:***

```
NetBIOS Support not in load.
```

2.15. NO

Use the **NO** command to delete information

Syntax:

```
ASRT config>NO ?
ADDRESS                     Add unique station address entries
BRIDGE                      Bridging functionality
DLS                         DLSw over the bridge
DUPLICATE                   Creation of duplicate frames in mixed environments
ETHERTYPE-IBMRT-PC          Translation of SNA frames to Ethernet 2 format
FA-GA-MAPPING               Group address to functional address (and vice versa)
IBM8209-SPANNING-TREE       Participate in spanning tree protocols with IBM 8209
MAPPING                     Functional address to group address mapping
PORT                        Adds a LAN/WAN port to the bridging configuration
PROT-FILTER                 Filter packets based on their protocol type
SOURCE-ROUTING              Source routing on a given port
SPANNING-TREE-EXPLORER      Port propagates spanning tree explorer frames
SR-TB-CONVERSION            Source-routing frame to transparent and vice versa
STP                         STP participation
TRANSPARENT                 Transparent bridging functionality on the given port
TREE                        STP participation for the bridge on a per-port basis
UB-ENCAPSULATION            Ungermann-Bass OUI encapsulation for XNS frames
```

a) NO ADDRESS

Deletes MAC address entry form from the permanent database.

Enter the add-value in 12-digit hexadecimal format. You cannot delete reserved multicast addresses. If you attempt to delete an address entry that does not exist, you receive a message like this

```
Record matching that address not Found
```

Example:

```
ASRT config>NO ADDRESS
Address (in 12-digit hex)[]? 001122334455
ASRT config>
```

b) NO MAPPING

Deletes specific address mapping for a given protocol.

dlh-type (Data-link-header type); Options are Destination Service Access Point (DSAP), Ethertype, or SNAP.

type-field Protocol type field.

Enter DSAP protocol type in a range of 1 to FE (hexadecimal).

Enter Ethernet (Ether) protocol type in a range of 5DD to FFF (hexadecimal).

Enter Subnetwork Access Protocol (SNAP) protocol type in 10-digit hexadecimal format.

ga-address 6-byte (12-digit hexadecimal) group/multicast address.

Syntax:

```
ASRT config>NO MAPPING ?
DSAP
ETHER
SNAP
```

· NO MAPPING DSAP

Example:

```
ASRT config>NO MAPPING DSAP
Protocol Type in hex (1 - FF)[1]? FE
Group-Address (in 12-digit hex)[]? AB0000020000
ASRT config>
```

· *NO MAPPING ETHER*

Example:

```
ASRT config>NO MAPPING ETHER
Protocol Type in hex (5DD - FFFF)[0800]?
Group-Address (in 12-digit hex)[]? ab0000020000
ASRT config>
```

· *NO MAPPING SNAP*

Example:

```
ASRT config>NO MAPPING SNAP
Address (in 10-digit hex)[0000000800]? 0000006002
Group-Address (in 12-digit hex)[]? AB0000020000
ASRT config>
```

c) <u>NO PROTOCOL-FILTER</u>

Deletes previously specified protocol identifiers used in filtering. You can delete filters for all ports or selected ports. These filters include the following:

SNAP Packets Subnetwork Access Protocol with protocol type entered in 10-digit hexadecimal format.

Ethernet Packets Ethernet type with the protocol type entered in a range of 5DD to FFFF (hexadecimal).

DSAP Packets Destination Service Access Point protocol with the protocol type entered in a range of 1 to FE (hexadecimal):

Syntax:

```
ASRT config>NO PROTOCOL-FILTER ?
DSAP
ETHER
SNAP
```

· *NO PROTOCOL-FILTER DSAP*

Example:

```
ASRT config>NO PROTOCOL-FILTER DSAP
Protocol Type in hex (1 - FE)[1]? 1
Port Number[1]?
ASRT config>
```

· *NO PROTOCOL-FILTER ETHER*

Example:

```
ASRT config>NO PROTOCOL-FILTER ETHER
Protocol Type in hex (5DD - FFFF)[800]? FFFF
Port Number[1]?
ASRT config>
```

· *NO PROTOCOL-FILTER SNAP*

Example:

```
ASRT config> NO PROTOCOL-FILTER SNAP
Address (in 10-digit hex)[0000000800]?
Port Number[1]?
ASRT config>
```

d) NO PORT

Removes a port from a bridging configuration. Since enable bridge by default configures all LAN devices to participate in bridging, this command allows you to specify which devices should or should not participate in bridging. The port number normally is one greater than the interface number. If port# is an IP tunnel, *port#* removes an IP tunnel from a bridging configuration.

Example:

```
ASRT config>NO PORT
Port Number[1]? 1
ASRT config>
```

e) NO BRIDGE

Completely deactivates the bridging functionality. This does not eliminate the previously configured bridging values.

Example:

```
ASRT config>NO BRIDGE
ASRT config>
```

f) NO DLS

Deactivates DLSw on the bridge. (The router running DLSw appears as a bridge to the end devices).

Example:

```
ASRT config>NO DLS
ASRT config>
```

g) NO DUPLICATE

Deactivates the creation of duplicate frames in mixed bridging environments. SR-TB on an 802.5 interface (with source-routing and transparent bridging enabled), may create inconsistencies when bridging frames to an unknown or multicast destination. The bridge does not know whether the destination is in a source-routing (only) or transparent bridge.

To remedy this, the bridge sends out duplicates of these frames (by default). One frame has source-routing fields (a spanning tree explorer RIF) and the other is formatted for transparent bridging (no RIF). The **NO DUPLICATE** command lets you eliminate this duplication by allowing you to disable the creation of one of these types of frames. The **NO DUPLICATE** command does not allow you to disable both types of frames simultaneously.

Entering **NO DUPLICATE STE** tells the bridge to refrain from sending spanning tree explorer frames created for the source-routing environment. Entering **NO DUPLICATE TSF** tells the bridge to refrain from sending out transparent spanning frames for the transparent bridging environment. In both cases, the bridge normally sends both types of frames. Disabling transparent bridging also disables the creation of transparent frames.

Syntax:

```
ASRT config>NO DUPLICATE ?
STE
TSF
```

· NO DUPLICATE STE

Example:

```
ASRT config>NO DUPLICATE STE
Port Number[1]? 1
ASRT config>
```

· *NO DUPLICATE TSF*

Example:

```
ASRT config>NO DUPLICATE TSF
Port Number[1]? 2
ASRT config>
```

h) NO ETHERTYPE-IBMRT-PC

Deactivates the translation of SNA frames to Ethernet 2 format used by the IBM RTs which execute OS/2/EE. For further information, please see chapter 5, section 2 “IBM RT characteristic for SNA Traffic”.

Example:

```
ASRT config>NO ETHERTYPE-IBMRT-PC
Port Number[1]? 1
ASRT config>
```

i) NO FA-GA-MAPPING

Deactivates group address to functional address (and vice versa) mapping. Under certain circumstances, you can disable the mapping between group address and functional address globally.

Example:

```
ASRT config>NO FA-GA-MAPPING
ASRT config>
```

j) NO IBM8209_SPANNING_TREE

Prevents the bridges from participating in the spanning tree protocols with IBM 8209 bridges.

Example:

```
ASRT config>NO IBM8209_SPANNING_TREE
ASRT config>
```

k) NO SOURCE-ROUTING

Disables source routing on a given port for an already participating bridge interface.

Example:

```
ASRT config>NO SOURCE-ROUTING
Port Number[1]?
ASRT config>
```

l) NO SR-TB-CONVERSION

Disables conversion of source-routing frame to transparent frame and vice versa.

Example:

```
ASRT config>NO SR-TB-CONVERSION
ASRT config>
```

m) NO STP

Deactivates STP participation for the entire bridge

Example:

```
ASRT config>NO STP
ASRT config>
```

n) NO SPANNING TREE-EXPLORER

Prevents a port from allowing propagation of spanning tree explorer frames if source routing is enabled. Use this command only if transparent bridging is *not* enabled on the port. In this case, it is automatically in conformance with the transparent spanning tree.

Example:

```
ASRT config>NO SPANNING-TREE-EXPLORER
Port Number[1]? 1
ASRT config>
```

o) NO TRANSPARENT

Disables transparent bridging functionality on the given port. This command is useful for cases where an alternative communication method such as source routing is desirable.

Also this command is useful when you want to enable, for example, SRB and SR-TB. But the command has pitfalls, so use it with care. For instance, using it on an Ethernet interface disables bridging for that interface. This command is used to enable SRB and SR-TB bridge functionality.

Example:

```
ASRT config>NO TRANSPARENT
Port Number[1]? 1
ASRT config>
```

p) NO TREE

Disables STP participation for the bridge on a per-port basis

Example:

```
ASRT config>NO TREE
Port Number[1]? 2
ASRT config>
```

Note: Deactivating STP on a per-port basis can produce loops in the network due to the existence of parallel bridges.

q) NO UB-ENCAPSULATION

Deactivates OUI Ungermann-Bass encapsulation for XNS frames. The bridge continues to transmit XNS frames to both Ethernet and Token Ring using SNAP encapsulation with an OUI set to all zeros as usual.

Example:

```
ASRT config>NO UB-ENCAPSULATION
ASRT config>
```

2.16. PORT

Adds a LAN/WAN port to the bridging configuration. Associates a port number with the interface number and enables that port's participation in transparent bridging.

Example:

```
ASRT config>PORT
Interface Name [ethernet0/0]? tnip1
Port Number[2]? 2
ASRT config>
```

If the interface is a Frame Relay interface, you are prompted to assign a circuit name.

Example:

```
ASRT config>PORT
Interface Name [ethernet0/0]? Fr1
Port Number[3]?
Assign circuit name[]? Prueba-01
ASRT config>
```

2.17. PROTOCOL-FILTER

Configures the bridge to filter packets based on their protocol type. It also discards matching ARP packets. The filters are applied to the selected ports.

The following protocol filters are available:

SNAP packets	Subnetwork Access Protocol with protocol type entered in 10-digit hexadecimal format.
Ether packets	Ethernet type with the protocol type entered in a range of 5DD to FFFF (Hexadecimal).
DSAP packets	DSAP protocol with the protocol type entered in a range of 1 to FE (hexadecimal).

You cannot add the enabled routing protocols to the router (the ones displayed by the **CONFIGURATION** command at the + prompt) for filtering. Common protocol filters and their values are as follows:

DSAP Types

Protocol	**SAP (hexadecimal value)**
Banyan SAP	BC (used only for 802.5)
Novell IPX SAP	EO (used only for 802.5)
NetBIOS SAP	FO
ISO Connectionless Internet	FE

SNAP Protocol Identifiers

Protocol	**SNAP OUILP (10-digit)**
AppleTalk Phase 2	08-00-07-80-9B
AppleARP Phase 2	00-00-00-80-F3
Proprietary	00-00-93-00-02
AppleTalk Phase 1 for FDDI	
Proprietary	00-00-93-00-03
AppleTalk ARP Phase 1 for FDDI	

Ethernet Types

Protocol	**Ethernet type (hexadecimal value)**
IP	0800
ARP	0806

CHAOS	0804
DECnet MOP Dump/Load	6000
DECnet MOP Remote Console	6002
DECnet	6003
DEC LAT	6004
DEC LAVC	6007
XNS	0600
Maintenance Packet Type	7030
Apollo Domain	8019 (Ethernet)
Novel NetWare IPX	8137 (Ethernet)
AppleTalk Phase 1	809B
AppleARP Phase 1	80F3
Loopback assistance	9000

Example 1:

```
ASRT config>PROTOCOL-FILTER DSAP
Protocol Type in hex (1 - FE)[1]?
Port Number[1]?
ASRT config>
```

Example 2:

```
ASRT config>PROTOCOL-FILTER ETHER
Protocol Type in hex (5DD - FFFF)[0800]?
Port Number[1]?
ASRT config>
```

Example 3:

```
ASRT config>PROTOCOL-FILTER SNAP
Address (in 10-digit hex)[0000000800]?
Port Number[1]?
ASRT config>
```

2.18. SET

Use the **SET** command to set the following parameters:

- Aging time for dynamic address entries in the filtering database
- Bridge address
- Largest Frame (LF) bit encoding interpretation for source routing
- MAC Service Data Unit (MSDU) size
- Spanning tree protocol bridge and port parameters
- Route Descriptor (RD) limit
- Size of the bridge filtering database

Syntax:

```
ASRT config>SET ?
AGE
BRIDGE
FILTERING
LF-BIT-INTERPRETATION
MAXIMUM-PACKET-SIZE
PORT
PROTOCOL
ROUTE-DESCRIPTOR-LIMIT
```

a) SET AGE

Sets the time for aging out dynamic entries in the filtering database when the port with the entry is in the forwarding state. This age is also used for aging RIF entries in the RIF table in the case of an SR-TB bridge personality.

The default for the aging timer is 300 seconds with a range of 1 to 1,000,000 seconds. The default for the resolution parameter is 5, with a range of 1 to 60 seconds.

Example:

```
ASRT config>SET AGE
seconds[300]? 250
resolution[5]?
ASRT config>
```

b) SET BRIDGE

Sets the bridge address. In cases where a serial line interface (or tunnel) is the lowest numbered port, you must use this command so that the bridge has a unique address when it is restarted. This is necessary because serial lines do not have their own MAC address.

Example:

```
ASRT config>SET BRIDGE
Bridge Address (in 12-digit hex)[]? 001122334455
ASRT config>
```

Note: Each bridge in the network must have a unique address for the spanning tree protocol to operate properly.

This is the low order 6-octet bridge address in the bridge identifier. By default, the *bridge-address* is set to the Media Access Control (MAC) address of the lowest numbered port at initialization. You can use this command to override the default address and enter your own unique address.

Do not use dashes or colons to separate each octet. If you enter the address in the wrong format you receive the message

```
Illegal Address
```

If you enter no address at the prompt, you receive the message

```
Zero length address supplied
```

and the bridge maintains its previous value. To return the bridge address to the default, enter an address of all zeroes.

c) SET FILTERING

Sets the number of entries that can be held in the bridge filtering database. The default is 1024 times the number of bridge ports. For more information see the **LIST FILTERING** command in this chapter.

Example:

```
ASRT config>SET FILTERING
database-size[2048]?
ASRT config>
```

d) SET LF-BIT-INTERPRETATION

Sets the Largest Frame (LF) bit encoding interpretation if source routing is enabled in this bridge.

Syntax:

```
ASRT config>SET LF-BIT-INTERPRETATION ?
BASIC
EXTENDED
```

· SET LF-BIT-INTERPRETATION BASIC

In **BASIC** mode only three bits of the routing control field are used. This is the common practice in source routing bridges that exist today. **EXTENDED** and **BASIC** nodes are compatible.

Example:

```
ASRT config>SET LF-BIT-INTERPRETATION BASIC
ASRT config>
```

· SET LF-BIT-INTERPRETATION EXTENDED

In **EXTENDED** mode, six bits of the routing control field are used to represent the maximum data unit that the bridge supports. The default is **EXTENDED**. **EXTENDED** and **BASIC** nodes are compatible.

Example:

```
ASRT config>SET LF-BIT-INTERPRETATION EXTENDED
ASRT config>
```

e) SET MAXIMUM-PACKET-SIZE

Sets the largest MAC Service Data Unit (MSDU) size for the port, if source routing is enabled on this port. Obviously, MSDU setting has no implication on traditionally transparent media. An MSDU value greater than the packet size configured in the router is treated as an error.

The default is the size configured as the packet size for that interface.

Example:

```
ASRT config>SET MAXIMUM-PACKET-SIZE
Port Number[1]? 2
MSDU size[4399]? 4000
MSDU is adjusted to 2052
ASRT config>
```

f) SET PORT

Permits you to enable or disable a bridge port.

Syntax:

```
ASRT config>SET PORT ?
BLOCK
DISABLE
```

· *SET PORT BLOCK*

Enables a port for those having bridge configured.

Example:

```
ASRT config>SET PORT BLOCK
Port Number[1]? 2
ASRT config>
```

· *SET PORT DISABLE*

Disables a port for those with bridge configured. The Port status passes to Disabled.

Example:

```
ASRT config>SET PORT DISABLE
Port Number[1]? 2
ASRT config>
```

g) SET PROTOCOL

Modifies the spanning tree protocol bridge or port parameters for a new configuration or to tune the configuration parameters to suit a specific topology.

· *SET PROTOCOL BRIDGE*

Enter **PROTOCOL BRIDGE** to modify with this command are described below.

When setting these values, make sure that the following relationships exist between the parameters or the input is rejected:

2 * (Bridge Forward Delay - 1 second) > Bridge Maximum Age

Bridge Maximum Age > 2 * (Bridge Hello Time + 1 second)

Example:

```
ASRT config>SET PROTOCOL BRIDGE
Bridge-Max-Age[20]? 25
Bridge-Hello-Time[2]?
Bridge-Forward-Delay[15]? 17
Bridge-Priority[32768]?
ASRT config>
```

Bridge Maximum Age Maximum age (period of time) used to time out spanning tree protocol related information.

Bridge Hello Time Time interval between Hello BPDUs.

Bridge Forward Delay Time interval before changing to another state (should this bridge become the root).

Bridge Priority A high-order 2-octet bridge address found in the Bridge Identifier - either the MAC address obtained from the lowest number port or the address set by the **SET BRIDGE** command.

· *SET PROTOCOL PORT*

Enter **PROTOCOL PORT** to modify the spanning tree protocol port parameters.

Example:

```
ASRT config>SET PROTOCOL PORT
Port Number[1]?
Port Path-cost (0 for default)[0]?
Default Path Cost is Selected
Port Priority[128]?
ASRT config>
```

Port Number Bridge port number; selects the port for which the path cost and port priority will be changed.

Port Path-cost Cost associated with the port which is used for possible root path cost. The range is 1 to 65535. 0 indicates the default path cost.

Port Priority Identifies port priority for the specified port. The range is 0 to 255.

h) SET ROUTE-DESCRIPTOR-LIMIT

Allows you to associate a maximum Route Descriptor (RD) length for All Route Explorer (ARE) or Spanning Tree Explorer (STE) frames forwarded by the bridge is source routing is enabled.

Syntax:

```
ASRT config>SET ROUTE-DESCRIPTOR-LIMIT ?
ARE
STE
```

· SET ROUTE-DESCRIPTION-LIMIT ARE

Entered as ARE depending on whether the *RD-limit-value* is applied to All Route Explorer (ARE).

Example:

```
ASRT config>SET ROUTE-DESCRIPTOR-LIMIT ARE
RD-limit-value (Hop count)[14]?
ASRT config>
```

· SET ROUTE-DESCRIPTION-LIMIT STE

Entered as STE depending on whether the *RD-limit-value* is applied to Spanning Tree Explorer (STE).

Example:

```
ASRT config>SET ROUTE-DESCRIPTOR-LIMIT STE 10
ASRT config>
```

RD-limit-value Specifies the maximum number of RDs that might be contained in the Routing Information Field (RIF) of the frame type specified by the RD limit type.

This field takes values from 0 to 14. The default RD limit value for ARE and STE frames is 14.

2.19. SOURCE-ROUTING

Enables source routing for a given port. Use this command when you want source routing on part of the bridge. If source routing is the only feature you want, disable transparent bridging on the interface. For the first instance of the command, you must type the bridge number. Subsequently, you need not.

port# Valid port participating in the bridge configuration.

segment# 12-bit number representing the LAN/WAN to which media are attached. All the media on other bridges attached to this LAN/WAN must be configured with the same value. For correct operations of source routing, it is very important that all the bridges attached to this LAN/WAN have the same perspective of the LAN/WAN identification value.

bridge# 4-bit value unique among all the bridges attached to the same LAN/WAN. This value is required when you enable source routing on the first interface. For later interfaces, this input is optional. It is recommended that *bridge#* be unique on the segment.

> ***Note: If the configuration is a situation where two segments have already been configured (i.e., a 1:N SRB configuration), you are prompted for an additional virtual-segment# parameter.***

Example:

```
ASRT config>SOURCE-ROUTING
Port Number[1]? 2
Segment Number for the port in hex(1 - FFF)[1]? 3
Bridge Virtual Segment Number in hex(1 - FFF)[1]? 2
ASRT config>
```

2.20. SPANNING-TREE-EXPLORER

Lets the port allow propagation of spanning tree explorer frames if source routing is enabled. This command is valid on Token Ring and WAN ports only. This feature is enabled by default when source routing is configured on the port.

Example:

```
ASRT config>SPANNING-TREE-EXPLORER
Port Number[1]? 2
ASRT config>
```

2.21. SR-TB-CONVERSION

Allows for compatibility between source routing and transparent bridging domains. When this feature is enabled, the bridge lets source-routed frames be accepted in a transparent domain by stripping off the RIF and converting them into transparent frames.

The bridge also gathers routing information concerning source routing stations from the RIFs of passing source-routing frames. It uses this RIF information to convert transparent frames to source-routed frames. If an RIF is not available for a station, then the bridge sends the frame out as a spanning tree explorer frame in the source-routing domain.

In order for the conversion to operate properly, you must give the transparent bridging domain a segment number. Configure SR-TB bridges connected to this domain with the same segment number.

Example:

```
ASRT config>SR-TB-CONVERSION
TB-Domain Segment Number in hex(1 - FFF)[1]?
Bridge Virtual Segment Number in hex(1 - FFF)[1]?
TB-Domain's MTU[1470]? 1400
TB-Domain's MTU is adjusted to 1350
ASRT config>
```

2.22. STP

Enables STP participation for the entire bridge.

Example:

```
ASRT config>STP
ASRT config>
```

2.23. TRANSPARENT

Enables transparent bridging functionality on the given port. Under normal circumstances, this command is not necessary.

Example:

```
ASRT config>TRANSPARENT
Port Number[1]? 2
ASRT config>
```

2.24. TREE

Enables STP participation for the bridge on a per-port basis.

Example:

```
ASRT config>TREE
Port Number[1]? 2
ASRT config>
```

2.25. UB-CAPSULATION

Causes XNS Ethernet 2 frames to be translated into Token Rings using the Ungermann-Bass OUI in the SNAP header. Forwards Token Ring frames containing the UB OUI header to Ethernets as type 0x0600 Ethernet 2 frames rather than as 802.3/802.2 frames.

Example:

```
ASRT config>UB-ENCAPSULATION
ASRT config>
```

2.26. EXIT

Use the **EXIT** command to return to the *Config>* prompt.

Syntax:

```
ASRT config>EXIT
```

Example:

```
ASRT config>EXIT
Config>
```

Chapter 8
ASRT Monitoring

1. ASRT Monitoring View

This section describes the ASRT monitoring commands. They allow you to specify parameters for the ASRT bridge and its interfaces. They also allow you to enable and monitor the NetBIOS.

To display the `ASRT>` monitoring prompt:

```
+PROTOCOL ASRT
ASRT>
```

Note: The bridge must be enabled in order to access the ASRT monitoring.

To access the NetBIOS monitoring commands, enter **NETBIOS** at the `ASRT>` prompt to get the `NetBIOS>` prompt.

```
ASRT>NETBIOS

NetBIOS Support User Console

NetBIOS>
```

2. ASRT Monitoring Commands

Command	**Function**
? (HELP)	Displays available commands.
ADD	Adds station addresses to the permanent database, address mapping, LAN/WAN ports, protocol filters, and a tunnel between end stations access an IP internetwork.
BAN	Displays the boundary access node (BAN) configuration or monitoring prompt.
CACHE	Displays cache entries for a specified port.
DELETE	Deletes station address entries, specific address mapping, LAN/WAN ports, protocol filters, and a tunnel between end stations across an IP internetwork.
FLIP	Flips MAC address from canonical to 802.5 (non-canonical or IBM) bit order.
LIST	Displays information about the complete bridge configuration or about selected configuration parameters.
NETBIOS	Displays the NetBIOS configuration or monitoring prompt.
NAME-CACHING	Permits enter into the Name Caching facility monitoring menu.
EXIT	Returns to the previous prompt.

2.1. ? (HELP)

List the commands available from the current prompt. After a specific command, lists its options.

Syntax:

```
ASRT>?
```

Example:

```
ASRT>?
ADD
BAN
CACHE port_number
DELETE mac_address
FLIP mac_address
LIST
NETBIOS
NAME-CACHING
EXIT
ASRT>
```

2.2. ADD

At the ASRT monitoring prompt, adds the following information to your bridging configuration. (These additions to the database do not survive after a restart).

Syntax:

```
ASRT>ADD ?
DESTINATION-ADDRESS-FILTER mac_address
STATIC-ENTRY mac_address input_port [output_ports...]
```

a) *ADD DESTINATION-ADDRESS-FILTER*

Adds a destination address filter to the router’s permanent database.

Example:

```
ASRT>ADD DESTINATION-ADDRESS-FILTER
Destination MAC address [00-00-00-00-00-00]? 00-01-02-03-04-05
ASRT>
```

b) *ADD STATIC-ENTRY*

Adds static address entries to the router’s permanent database. The output ports are optional.

To create a static entry with multiple port maps (one per input port), enter the command several times.

Example:

```
ASRT>ADD STATIC-ENTRY
MAC address [00-00-00-00-00-00]? 11-22-33-44-55-66
input port, 0 for any[0]?
output port, 0 for none[0]? 1
output port, 0 to end[0]? 0
ASRT>
```

2.3. BAN

Displays the BAN> prompt. You can access this prompt by entering **BAN** at the ASRT> prompt as shown below.

Syntax:

```
ASRT>BAN
```

Example:

```
ASRT>BAN
Boundary Access Node Console
BAN>
```

2.4. CACHE

Displays the contents of a selected bridging port routing cache. If the port does not have a cache, you see the message

```
PORT DOESN’T HAVE A CACHE
```

Syntax:

```
ASRT>CACHE <port#>
```

Example:

```
ASRT>CACHE
Port Number[1]? 2
MAC Address    MC*  Entry Type       Age  Port(s)
00-00-93-00-c0-d0   Dynamic           20  2 (TKR/1)
ASRT>
```

MAC Address 6-byte MAC address of the entry.

Entry Type Displays one of the following address entry types:

Reserved Reserved by the IEEE802.1D Standard.

Registered Unicast addresses belonging to proprietary communications hardware attached to the multicast addresses enabled by protocol forwarders.

Permanent User configured entries.

Static Monitoring entries.

Dynamic Learned by the bridge dynamically. Do not survive power on/offs or system resets and have an age associated with the entry.

Free Locations in database that are free to be filled by address entries.

Unknown Unknown to the bridge. May be bugs and/or illegal addresses

Age Age in seconds of each dynamic entry. Age is decremented at each resolution intervals.

Port(s) The port number associated with the entry. Displays the interface name (always that of the interface having the cache).

2.5. DELETE

Deletes station (MAC) address entries from the permanent database.

Syntax:

```
ASRT>DELETE <MAC address>
```

Example:

```
ASRT>DELETE
MAC address [00-00-00-00-00-00]? 00-01-02-03-04-05
ASRT>
```

2.6. FLIP

Lets you view specific MAC addresses in the canonical and non-canonical formats by flipping the address bit order. **FLIP** translates IEEE 802.5 addresses in their typical non-canonical format to the canonical format universally used by the bridge monitoring process and ELS and vice versa.

Syntax:

```
ASRT>FLIP <MAC address>
```

Example:

```
ASRT>FLIP
MAC address [00-00-00-00-00-00]? 01-02-03-04-05-06
IEEE 802 canonical bit order:    01-02-03-04-05-06
IBM Token-Ring native bit order: 80:40:c0:20:a0:60

ASRT>
```

2.7. LIST

Displays information about the complete bridge configuration or about selected configuration parameters.

Syntax:

```
ASRT>LIST ?
ADAPTIVE
BRIDGE
CONVERSION
DATABASE
FILTERING
PORT
SOURCE-ROUTING
SPANNING-TREE-PROTOCOL
TRANSPARENT
```

a) LIST ADAPTIVE

Lists all general information regarding the SR-TB bridge which converts between types of bridging. There are a number of general data group options that may be displayed under the **LIST ADAPTIVE**. These include the following:

- *Config* - Displays general information regarding the SR-TB bridge.
- *Counters* - Displays all SR-TB bridge counters.
- *Database* - Displays contents of the SR-TB bridge RIF database.

Syntax:

```
ASRT config>LIST ADAPTIVE ?
CONFIG
COUNTERS
DATABASE
```

· LIST ADAPTIVE CONFIG

Example:

```
ASRT>LIST ADAPTIVE CONFIG
Adaptive bridge:            Enabled
Translation database size:  0
Aging time:                 15 seconds
Aging granularity           5 seconds

Port  Segment  Interface      State       MTU  DUP:TSF STE
   1  001      TKR/0          Enabled     2052      Yes   Yes
   -  001      Adaptive       Enabled     1470
ASRT>
```

Adaptive bridge Current state of the SR-TB adaptive bridge, either enabled or disabled.

Translation database size Current size of the SR-TB database, which contains MAC addresses and associated RIFs for the source-routing domain.

Aging time Aging timer setting in seconds. All SR:TB RIF database entries that exceed this time limit are discarded.

Aging granularity How often entries are scanned to look for expiration according to the aging timer.

Port Number of a port associated with conversion bridging.

Segment Source-routing segment number assigned to the port associated with conversion bridging.

Interface Device connected to a conversion bridge network segment.

State Current state of the conversion bridge port.

MTU Maximum frame size (from the end of the RIF to the beginning of the FCS) that the conversion-bridge can transmit and receive.

DUP: TSF STE Indicates if duplicated STE (Spanning Tree Explorer) or TSF (Transparent Spanning Frames) frames are sent.

· *LIST ADAPTIVE COUNTERS*

Example:

```
ASRT>LIST ADAPTIVE COUNTERS
Hash collision count:                 0
Adaptive database overflow count:     0
ASRT>
```

Hash Collision Count Number of addresses that were stored (hashed) to the same location in the hash table. This number is cumulative and reflects the total number of hash collision incidents that occurred. Increases in this number may indicate a potential table size problem.

Adaptive Database Overflow Number of times that an address was overwritten as the conversion database table ran out of table space.

· *LIST ADAPTIVE DATABASE*

The **LIST ADAPTIVE DATABASE** command lets you select certain portions of the adaptive bridge RIF database to display. This is due to the potential size of the database. The display options include the following:

ADDRESS - Displays data on the address found in the database.

ALL-SEGMENTS- Displays the entire database.

PORT - Displays all conversion bridge entries for a specific port.

SEGMENT - Displays all conversion bridge entries associated with the port having the specified segment number.

The following example illustrates each of the list adaptive-bridge database command options

> ***Note: These are only displayed if adaptive bridging is enabled.***

Example 1:

```
ASRT>LIST ADAPTIVE DATABASE ADDRESS
MAC address [00-00-00-00-00-00]? 00a026400ba4
Canonical MAC address:           00-a0-26-40-0b-a4
IBM Token-Ring native address:   80:05:64:02:d0:25
Via port:                        1 (TKR/0       )
Entry age:                       315
RIF Routing type:                ARE (100)
RIF length:                      6
RIF Direction:                   1
RIF Largest frame:               1470
RIF Route Descriptor  LAN ID     Bridge Number
1                     100        1
2                     200        0
ASRT>
```

Example 2:

```
ASRT>LIST ADAPTIVE DATABASE ALL-SEGMENTS
Canonical Address  Interface    Port  Seg    Age  RIF: Type Direct   Length  LF
IBM MAC Address    RIF

00-00-93-78-b7-3a  TKR/0           1  100    310       ARE  Reverse   6      1470
80:00:c9:1e:ed:5c  869010012000

00-a0-26-40-0b-a4  TKR/0           1  100    320       ARE  Reverse   6      1470
80:05:64:02:d0:25  869010012000

ASRT>
```

Example 3:

```
ASRT>LIST ADAPTIVE DATABASE PORT
Port number[1]? 2
Canonical Address  Interface    Port  Seg    Age  RIF: Type Direct   Length  LF
IBM MAC Address    RIF

00-0a-83-78-b7-a4  TKR/1           2  200    300       ARE  Reverse   6      1470
80:00:c9:1e:ed:25  869010011000

ASRT>
```

Example 4:

```
ASRT>LIST ADAPTIVE DATABASE SEGMENT
Segment number[1]? 100
Canonical Address  Interface    Port  Seg    Age  RIF: Type Direct   Length  LF
IBM MAC Address    RIF

00-00-93-78-b7-3a  TKR/0           1  100    315       ARE  Reverse   6      1470
80:00:c9:1e:ed:5c  869010012000

00-a0-26-40-0b-a4  TKR/0           1  100    320       ARE  Reverse   6      1470
80:05:64:02:d0:25  869010012000

ASRT>
```

The following information is displayed for each entry:

Canonical address MAC address of the node corresponding to this entry displayed in the IBM non-canonical bit order.

Interface Name of the network interface that learned this entry.

Port Number of the port that learned this address entry.

Seg Number of the segment that learned this address.

Age Entry age in seconds.

RIF Type RIF type (SRF, STE, or ARE).

RIF Direction RIF direction (Forward or Reverse).

RIF Length RIF length in bytes.

RIF LF Largest frame value in the RIF.

RIF RIF (Routing Information Field) learned from this node.

b) LIST BRIDGE

Lists all general information regarding the bridge.

Example:

```
ASRT>LIST BRIDGE
Bridge ID (prio/add):  32768/00-a0-26-40-0c-e4
Bridge state:          Enabled
UB-Encapsulation:      Disabled
Bridge type:           SR-TB
Bridge capability:     ASRT
Number of ports:       2
STP Participation:     IEEE802.1d on TB ports and IBM-8209 on SR ports

                                                           Maximum
Port   Interface      State     MAC Address          Modes  MSDU  Segment  Flags
   1   TKR/0           Up        00-a0-26-40-0c-e4    SR     2096     100     RD
   2   Eth/0           Up        00-a0-26-40-0c-e5       T  1514              RD

Flags:  RE = IBMRT PC behavior Enabled,  RD = IBMRT PC behavior Disabled

SR bridge number:      1
SR virtual segment:    000
Adaptive segment:      200
ASRT>
```

Bridge ID (prio/add) Bridge identifier.

Bridge State Indicates whether bridging is enabled or disabled.

UB-Encapsulation Indicates if the UB encapsulation is enabled.

Bridge Type The configured bridge type (None, SRB, STB, SRT, SR-TB or ASRT).

Bridge capability Bridge capacity (ASRT, STB, SRB or STB/SRB).

Number of Ports Number of ports configured for that bridge.

STP Participation Participation type in the Spanning Tree Protocol.

Port Number assigned to an interface using the **PORT** command.

Interface Devices connected to a network segment through the bridge.

State The current state of the port (Up or Down).

MAC address The MAC address associated with that port in canonical bit order.

Modes The bridging mode for that port. *T* indicates transparent bridging. *SR* indicates source routing. *A* indicates adaptive bridging.

MSDU The maximum frame (data unit) size (including the MAC header but not the FCS field) the source-routing bridge can transmit and receive on this interface.

Segment The source routing bridge segment number assigned to that port (if any).

FLAGS Indicates if the IBM RT is enabled.

SR bridge number The user-assigned source routing bridge number.

SR virtual segment The source-routing bridge virtual segment number, if any.

Adaptive segment The number of the segment used in the source-routing domain to route to the transparent domain.

c) LIST CONVERSION

Displays general information about the bridge’s rules for converting frame formats based on the frame type. You can display the following general data groups under the **LIST CONVERSION** command.

Syntax:

```
ASRT>LIST CONVERSION ?
ALL
ETHERTYPE
SAP
SNAP
```

· *LIST CONVERSION ALL*

Displays all rules.

Example:

```
ASRT>LIST CONVERSION ALL
Ethernet type 0800 translations:
Group ab-00-00-04-00-00 <=> Functional c0-00-08-00-00-00 (03:00:10:00:00:00)

IEEE 802.2 destination SAP 01 translations:
Group ab-00-00-01-00-00 <=> Functional c0-00-30-00-00-00 (03:00:0c:00:00:00)

IEEE 802 SNAP PID 00-00-00-60-02 translations:
Group ab-00-00-02-00-00 <=> Functional c0-00-20-00-00-00 (03:00:04:00:00:00)

ASRT>
```

· *LIST CONVERSION ETHERTYPE*

Displays rules for all Ethernet types or for a specific Ethernet type.

Example:

```
ASRT>LIST CONVERSION ETHERTYPE
Ethernet type (in hexadecimal), 0 for all[0]?
Ethernet type 0800 translations:
Group ab-00-00-04-00-00 <=> Functional c0-00-08-00-00-00 (03:00:10:00:00:00)

ASRT>
```

· *LIST CONVERSION SAP*

Displays rules for all SAP protocol identifiers or a specific 802.2 SAP type.

Example:

```
ASRT>LIST CONVERSION SAP
SAP (in hexadecimal), 100 for all[100]?
IEEE 802.2 destination SAP 01 translations:
Group ab-00-00-01-00-00 <=> Functional c0-00-30-00-00-00 (03:00:0c:00:00:00)

ASRT>
```

· *LIST CONVERSION SNAP*

Displays rules for all SNAP protocol identifiers or a specific 802.2 SNAP type.

Example:

```
ASRT>LIST CONVERSION SNAP
SNAP Protocol ID, return for all [00-00-00-00-00]?
IEEE 802 SNAP PID 00-00-00-60-02 translations:
Group ab-00-00-02-00-00 <=> Functional c0-00-20-00-00-00 (03:00:04:00:00:00)

ASRT>
```

d) LIST DATABASE

Lists the contents of transparent filtering databases. You can choose the following data groups to display under the **LIST DATABASE** command.

Syntax:

```
ASRT>LIST DATABASE ?
ALL-PORTS
DYNAMIC
LOCAL
PERMANENT
PORT port_number
RANGE mac_address mac_address
STATIC
```

· *LIST DATABASE ALL-PORTS*

Displays the entire transparent bridging database.

Example:

```
ASRT>LIST DATABASE ALL
MAC Address    MC*  Entry Type       Age  Port(s)

00-00-0c-07-ac-00   Dynamic          320  2 (Eth/0       )
00-00-0c-07-ac-0d   Dynamic          320  2 (Eth/0       )
00-00-24-31-33-c1   Dynamic          315  2 (Eth/0       )
00-00-b4-95-33-bc   Dynamic          280  2 (Eth/0       )
00-00-c0-57-eb-6b   Dynamic          275  2 (Eth/0       )
00-00-c0-6a-eb-6b   Dynamic          120  2 (Eth/0       )
00-60-08-79-33-4d   Dynamic          315  2 (Eth/0       )
00-60-08-79-33-5a   Dynamic          315  2 (Eth/0       )
00-60-52-02-83-42   Dynamic          130  2 (Eth/0       )
00-60-97-13-8c-cd   Dynamic          220  2 (Eth/0       )
00-60-97-13-8d-77   Dynamic          235  2 (Eth/0       )
00-60-97-27-bc-c6   Dynamic          315  2 (Eth/0       )
00-60-97-3c-48-f4   Dynamic          250  2 (Eth/0       )
00-60-97-3e-52-d5   Dynamic          235  2 (Eth/0       )
00-60-97-3e-53-cb   Dynamic          290  2 (Eth/0       )
00-60-97-3e-53-d6   Dynamic           10  2 (Eth/0       )
00-60-97-3e-53-d7   Dynamic          320  2 (Eth/0       )
00-60-97-3e-6c-07   Dynamic          315  2 (Eth/0       )
00-60-97-3e-6d-a7   Dynamic           75  2 (Eth/0       )
00-80-5f-43-bc-ec   Dynamic          190  2 (Eth/0       )
00-80-5f-a6-04-d0   Dynamic          295  2 (Eth/0       )
00-a0-24-23-d3-8a   Dynamic          110  2 (Eth/0       )
00-a0-26-40-0e-ec   Dynamic          320  2 (Eth/0       )
00-a0-26-40-10-ac   Dynamic          285  2 (Eth/0       )
00-a0-26-5c-4e-26   Dynamic          195  2 (Eth/0       )
00-a0-c9-a7-e6-66   Dynamic          320  2 (Eth/0       )
00-b0-64-b8-8b-40   Dynamic          300  2 (Eth/0       )
00-c0-4f-0a-37-f2   Dynamic          320  2 (Eth/0       )
00-c0-4f-9c-11-b7   Dynamic          230  2 (Eth/0       )
00-c0-4f-9c-12-2a   Dynamic          310  2 (Eth/0       )
00-d0-58-35-d2-e0   Dynamic          270  2 (Eth/0       )
01-80-c2-00-00-00*  Registered            1
01-80-c2-00-00-01*  Reserved              All
01-80-c2-00-00-02*  Reserved              All
01-80-c2-00-00-03*  Reserved              All
01-80-c2-00-00-04*  Reserved              All
01-80-c2-00-00-05*  Reserved              All
01-80-c2-00-00-06*  Reserved              All
01-80-c2-00-00-07*  Reserved              All
01-80-c2-00-00-08*  Reserved              All
01-80-c2-00-00-09*  Reserved              All
01-80-c2-00-00-0a*  Reserved              All
01-80-c2-00-00-0b*  Reserved              All
01-80-c2-00-00-0c*  Reserved              All
01-80-c2-00-00-0d*  Reserved              All
01-80-c2-00-00-0e*  Reserved              All
01-80-c2-00-00-0f*  Reserved              All
03-00-00-00-80-00*  Reserved              All
08-00-09-a3-04-21   Dynamic          270  2 (Eth/0       )
08-00-20-83-56-ff   Dynamic          320  2 (Eth/0       )
08-00-4e-09-ba-4c   Dynamic          320  2 (Eth/0       )
```

```
08-00-4e-12-da-34   Dynamic          205  2 (Eth/0       )
08-00-5a-93-6d-fa   Dynamic          305  2 (Eth/0       )

ASRT>
```

Note: The fields described below are displayed for all of the list database command options.

MAC Address Address in 12-digit hex format (canonical bit order).

*MC** An asterisk following an address entry indicates that the entry has been flagged as a multicast address.

Entry Type Specifies one of the following types:

Reserved Reserved by the IEEE802.1D standard.

Registered Consists of unicast addresses belonging to interfaces participating in the bridge or multicast addresses enabled by protocol forwarders.

Permanent Entered in the configuration process and survives power on/offs or system resets.

Static Entered in the monitoring process, do not survive power on/offs or system resets and are ageless.

Dynamic Learned by the bridge dynamically and do not survive power on/offs or system resets and have an age associated with the entry.

Free This type is not used and should not be seen except in occasional *race* conditions between the monitor process and the bridge.

Unknown Unknown entry type. May indicate a software bug. Report the hex entry type to Customer Service.

Age The age (in seconds) of each dynamic entry. Age is decremented at each resolution interval

Port(s) The outgoing port number(s) for that entry. Device type is also listed for single port entries. If dynamic entry on IP tunnel, the port is 5 for IP tunnel.

· *LIST DATABASE DYNAMIC*

Displays all dynamic (learned) address database entries.

Example:

```
ASRT>LIST DATABASE DYNAMIC
MAC Address    MC*  Entry Type       Age  Port(s)

00-00-0c-07-ac-00   Dynamic          315  2 (Eth/0       )
00-00-0c-07-ac-0d   Dynamic          315  2 (Eth/0       )
00-00-24-31-33-c1   Dynamic          140  2 (Eth/0       )
00-00-b4-95-33-bc   Dynamic          250  2 (Eth/0       )
00-00-c0-57-eb-6b   Dynamic          175  2 (Eth/0       )
00-00-c0-6a-eb-6b   Dynamic           50  2 (Eth/0       )
00-00-e8-41-ad-13   Dynamic          315  2 (Eth/0       )
ASRT>
```

· LIST DATABASE LOCAL

Displays all local (reserved) address database entries.

Example:

```
ASRT>LIST DATABASE LOCAL
MAC Address    MC*  Entry Type       Age  Port(s)

00-a0-26-40-0c-e4   Registered            1 (TKR/0       )
00-a0-26-40-0c-e5   Registered            2 (Eth/0       )
01-80-c2-00-00-00*  Registered            1
ASRT>
```

· LIST DATABASE PERMANENT

Displays all permanent address database entries.

Example:

```
ASRT>LIST DATABASE PERMANENT
MAC Address    MC*  Entry Type       Age  Port(s)

00-11-22-33-44-55   Permament             1 (TKR/0       )  ->  1-2
ASRT>
```

· LIST DATABASE PORT

All entries falling within this range are displayed.

Example:

```
ASRT>LIST DATABASE PORT
Port Number[1]?
MAC Address    MC*  Entry Type       Age  Port(s)

00-a0-26-40-0c-e4   Registered            1 (TKR/0       )
01-80-c2-00-00-00*  Registered            1
01-80-c2-00-00-01*  Reserved              All
01-80-c2-00-00-02*  Reserved              All
01-80-c2-00-00-03*  Reserved              All
01-80-c2-00-00-04*  Reserved              All
01-80-c2-00-00-05*  Reserved              All
01-80-c2-00-00-06*  Reserved              All
01-80-c2-00-00-07*  Reserved              All
01-80-c2-00-00-08*  Reserved              All
01-80-c2-00-00-09*  Reserved              All
01-80-c2-00-00-0a*  Reserved              All
01-80-c2-00-00-0b*  Reserved              All
01-80-c2-00-00-0c*  Reserved              All
01-80-c2-00-00-0d*  Reserved              All
01-80-c2-00-00-0e*  Reserved              All
01-80-c2-00-00-0f*  Reserved              All
03-00-00-00-80-00*  Reserved              All
ASRT>
```

· LIST DATABASE RANGE

Displays a range of database entries from the total transparent bridging filtering address database. A starting and stop MAC address is given to define the range. All entries that are within this range are displayed.

Example:

```
ASRT> LIST DATABASE RANGE
First MAC address [00-00-00-00-00-00]?
Last MAC address [FF-FF-FF-FF-FF-FF]? 00-00-ff-ff-ff-ff
```

```
MAC Address    MC*  Entry Type       Age  Port(s)

00-00-0c-07-ac-00   Dynamic          315  2 (Eth/0       )
00-00-0c-07-ac-0d   Dynamic          315  2 (Eth/0       )
00-00-b4-95-33-bc   Dynamic          270  2 (Eth/0       )
00-00-c0-57-eb-6b   Dynamic          315  2 (Eth/0       )
00-00-c0-eb-51-a4   Dynamic          290  2 (Eth/0       )
00-00-e8-3d-1b-bc   Dynamic          240  2 (Eth/0       )
00-00-e8-3d-1c-dc   Dynamic          305  2 (Eth/0       )
00-00-e8-3d-31-48   Dynamic          275  2 (Eth/0       )
00-00-e8-3d-a5-04   Dynamic          270  2 (Eth/0       )
00-00-e8-41-ad-13   Dynamic          265  2 (Eth/0       )
ASRT>
```

· *LIST DATABASE STATIC*

Displays static entries from the address database.

Example:

```
ASRT>LIST DATABASE STATIC
MAC Address    MC*  Entry Type       Age  Port(s)

01-02-03-0a-0b-0c*  Static                1 (TKR/0       )  ->  1-2
ASRT>
```

e) LIST FILTERING

You can display the following general data group under the **LIST FILTERING** command.

Syntax:

```
ASRT>LIST FILTERING ?
ALL
ETHERTYPE
SAP
SNAP
```

· *LIST FILTERING ALL*

Displays all filtering database entries.

Example:

```
ASRT>LIST FILTERING ALL
Ethernet type 9000 is bridged & processed on ports 1-2
IEEE 802.2 destination SAP 00 is bridged & processed on ports 1-2
IEEE 802.2 destination SAP 42 is routed on ports 1-2
IEEE 802 SNAP PID 00-00-00-90-00 is bridged & processed on ports 1-2
ASRT>
```

Descriptors used to explain how packets are communicated include the following:

- Routed - Packets that are passed to routing forwarder to be forwarded.
- Filtered - Packets that are administratively filtered by user setting protocol filters.
- Bridged and routed - A protocol identifier for which there is a protocol entity within the system that is not a forwarder. An example is a link-level echo protocol. Unicast packets from this protocol are bridged or locally processed if being sent to a registered address. Multicast packets are forwarded and locally processed for a registered multicast address.

All of the descriptors just explained also apply to ARP packets with this Ethertype.

· *LIST FILTERING ETHERTYPE*

Displays Ethernet protocol type filter database entries.

Example:

```
ASRT>LIST FILTERING ETHERTYPE
Ethernet type (in hexadecimal), 0 for all[0]?
Ethernet type 9000 is bridged & processed on ports 1-2
ASRT>
```

· *LIST FILTERING SAP*

Displays SAP protocol filter database entries.

Example:

```
ASRT>LIST FILTERING SAP
SAP (in hexadecimal), 100 for all[100]?
IEEE 802.2 destination SAP 00 is bridged & processed on ports 1-2
IEEE 802.2 destination SAP 42 is routed on ports 1-2
ASRT>
```

· *LIST FILTERING SNAP*

Displays SNAP protocol identifier filter database entries.

Example:

```
ASRT>LIST FILTERING SNAP
SNAP Protocol ID, return for all [00-00-00-00-00]?
IEEE 802 SNAP PID 00-00-00-90-00 is bridged & processed on ports 1-2
ASRT>
```

f) LIST PORT

Displays the status of the bridge ports.

Example:

```
ASRT>LIST PORT
Port Number[-1]?
Port Id (dec)    : 128: 1, (hex): 80-01
Port State       : Forwarding
STP Participation: Enabled
Port Supports    : Source Routing Bridging Only
SRB: Segment Number: 0x100       MTU:  2052     STE Forwarding: Auto
Assoc Interface #/name : 00/TKR/0
-----------------------------------------------------------------------------
Port Id (dec)    : 128: 2, (hex): 80-02
Port State       : Forwarding
STP Participation: Enabled
Port Supports    : Transparent Bridging Only
Duplicates Frames Allowed:   STE: Yes  , TSF: Yes
Assoc Interface #/name : 01/Eth/0
-----------------------------------------------------------------------------
ASRT>
```

g) LIST SOURCE ROUTING

Displays source-routing bridge configuration information. There are general data group options that you can display under the **LIST SOURCE-ROUTING** command:

Syntax:

```
ASRT>LIST SOURCE-ROUTING ?
CONFIGURATION
COUNTERS
STATE
```

· *LIST SOURCE-ROUTING CONFIGURATION*

Displays general information regarding the SRB bridge.

Example:

```
ASRT>LIST SOURCE-ROUTING CONFIGURATION
Bridge number:               1
Bridge state:                Enabled
Maximum STE hop count        14
Maximum ARE hop count        14
Virtual segment:             000

Port   Segment    Interface      State       MTU  STE Forwarding
  1        100    TKR/0          Enabled    2052      Auto
  -        200   Adaptive       Enabled   1470        Yes
ASRT>
```

Bridge number Bridge number (in hexadecimal) assigned to this bridge.

Bridge state Indicates whether bridging is enabled or disabled.

Maximum STE hop count Maximum hop count for Spanning Tree Explorer frames transmitting from the bridge for a given interface associated with source routing bridging.

Maximum ARE hop count Maximum hop count for All Route Explorer frames transmitting from the bridge for a given interface associated with source routing bridging.

Virtual segment Virtual segment number assigned for 1:N bridging.

Port Numbers of ports associated with source routing bridging.

Segment Assigned segment numbers for networks associated with source routing bridging.

Interface Associated interface names. Lists Adaptive for interfaces participating in the SR-TB.

State Current port state (Enabled or Disabled).

MTU MTU size set for that port.

STE Forwarding Indicates whether Spanning Tree Explorers received on this port are forwarded (Yes) and whether STEs from other ports go out this port.

· *LIST SOURCE-ROUTING COUNTERS*

Displays all SRB bridge counters.

Syntax:

```
ASRT>LIST SOURCE-ROUTING COUNTERS ?
ALL-PORTS
PORT port_number
SEGMENT segment_number
```

The counters option has further subgroups of information that you can display using the **LIST SOURCE-ROUTING** command:

- *All-ports* - Displays counters for all ports.
- *Port* - Displays counters for a specific port.
- *Segment* - Displays counters for the port corresponding to a specific segment.

The following examples illustrate each of the **LIST SOURCE-ROUTING** display options.

Example 1:

```
ASRT>LIST SOURCE-ROUTING COUNTERS ALL
Counters for port 1, segment 100, interface TKR/0       :
SRF frames received:          0    sent:          0
STE frames received:      18876    sent:          0
ARE frames received:        168    sent:          0
SR frames sent as TB:                        0
TB frames sent as SR:                    26494
Dropped, in queue overflow:                  0
Dropped, source address filter:              0
Dropped, destination address filter:         0
Dropped, protocol filtering:                 0
Dropped, invalid ri length:                  0
Dropped, duplicated segment:             18814
Dropped, segment mismatch:                   0
Dropped, duplicated lan id:                  0
Dropped, stehop count exceeded:              0
Dropped, arehop count exceeded:              0
Dropped, no buffer available:                0
Dropped, mtu exceeded:                       0

Counter for port - segment 200, Adaptive:
ASRT>
```

Port Numbers of ports associated with source routing bridging.

Segment Source-routing segment numbers in hex.

Interface Name of the network interface.

SRF Frames Received/Sent Specifically Routed Frames received or sent on this bridge.

STE Frames Received/Sent Spanning Tree Explorer Frames received or sent on this bridge.

ARE Frames Received/Sent All Routes Explorer Frames received or sent on this bridge.

SR Frames Sent as TB Source routing frames received on this interface that were sent as TransparentBridge frames.

TB Frames Sent as SR Transparent bridge frames received on this interface that were sent as source routing frames.

Dropped, in queue overflow Frames dropped because the input queue overflowed.

Dropped, source address filter Frames dropped because this source address matched a source address filter in the filtering database.

Dropped, destination address filter Frames dropped because this destination address matched a source address filter in the filtering database.

Dropped, protocol filtering Frames dropped because their protocol identifier is being administratively filtered.

Dropped, invalid ri length Frames dropped because the RIF length was less than 2 or over 30.

Dropped, duplicate segment Frames dropped because of a duplicate segment in the RIF. This is normal for the ARE frames.

Dropped, segment mismatch Frames dropped because the outgoing segment number does not match any in this bridge.

Dropped, duplicated lan id Frames discarded due to a duplicated LAN ID.

Dropped, stehop count exceeded Frames discarded because the STE has surpassed the number of permitted hops.

Dropped, arehop count exceeded Frames discarded because the ARE has surpassed the number of permitted hops.

Dropped, no buffer available Frames discarded as there is no buffer available.

Dropped, mtu exceeded Frames discarded as the MTU has been exceeded.

Example 2:

```
ASRT>LIST SOURCE-ROUTING COUNTERS PORT
Port Number[1]?
Counters for port 1, segment 100, interface TKR/0       :
SRF frames received:          0    sent:          0
STE frames received:      25134    sent:          0
ARE frames received:        231    sent:          0
SR frames sent as TB:                        0
TB frames sent as SR:                    35349
Dropped, in queue overflow:                  0
Dropped, source address filter:              0
Dropped, destination address filter:         0
Dropped, protocol filtering:                 0
Dropped, invalid ri length:                  0
Dropped, duplicated segment:             25048
Dropped, segment mismatch:                   0
Dropped, duplicated lan id:                  0
Dropped, stehop count exceeded:              0
Dropped, arehop count exceeded:              0
Dropped, no buffer available:                0
Dropped, mtu exceeded:                       0

ASRT>
```

Example 3:

```
ASRT>LIST SOURCE-ROUTING COUNTERS SEGMENT
Segment number[1]? 100
Counters for port 1, segment 100, interface TKR/0       :
SRF frames received:          0    sent:          0
STE frames received:      25285    sent:          0
ARE frames received:        232    sent:          0
SR frames sent as TB:                        0
TB frames sent as SR:                    35570
Dropped, in queue overflow:                  0
Dropped, source address filter:              0
Dropped, destination address filter:         0
Dropped, protocol filtering:                 0
Dropped, invalid ri length:                  0
Dropped, duplicated segment:             25198
Dropped, segment mismatch:                   0
Dropped, duplicated lan id:                  0
Dropped, stehop count exceeded:              0
Dropped, arehop count exceeded:              0
Dropped, no buffer available:                0
Dropped, mtu exceeded:                       0

ASRT>
```

· *LIST SOURCE-ROUTING STATE*

Displays information related to the SRB bridge status.

Example:

```
ASRT>LIST SOURCE-ROUTING STATE

Bridge state:                Up

Port   Segment    Interface      State  STE Forwarding
   1       100    TKR/0          Up          Yes

ASRT>
```

h) LIST SPANNING-TREE-PROTOCOL

Displays spanning tree protocol information. The transparent bridge uses the spanning tree protocol to form a loop-free topology. You can display the following general data group options under the **LIST SPANNING-TREE-PROTOCOL** command:

Syntax:

```
ASRT>LIST SPANNING-TREE-PROTOCOL ?
CONFIGURATION
COUNTERS
STATE
TREE
```

· LIST SPANNING-TREE-PROTOCOL CONFIGURATION

Displays information concerning the spanning tree protocol.

Example:

```
ASRT>LIST SPANNING-TREE-PROTOCOL CONFIGURATION
Bridge ID (prio/add):  32768/00-a0-26-40-0c-e4
Bridge state:          Enabled
Maximum age:           20 seconds
Hello time:            2 seconds
Forward delay:         15 seconds
Hold time:             1 seconds
Filtering age:         320 seconds
Filtering resolution:  5 seconds

Port  Interface      Priority      Cost      State
   1  TKR/0               128      1062      Enabled
   2  Eth/0               128       100      Enabled
ASRT>
```

· LIST SPANNING-TREE-PROTOCOL COUNTERS

Displays the spanning tree protocol counters.

Example:

```
ASRT>LIST SPANNING-TREE-PROTOCOL COUNTERS
Time since tolopology change (seconds)       0
Topolgy changes:                             4
BPDUs received:                              1
BPDUs sent:                                  5673

Port  Interface      BPDUs received  BDPU input overflow  Forward transitions
   1    TKR/0                     0                    0                 1
   2    Eth/0                     1                    0                 1
ASRT>
```

· LIST SPANNING-TREE-PROTOCOL STATE

Displays the current spanning tree protocol state information.

Example:

```
ASRT>LIST SPANNING-TREE-PROTOCOL STATE
Designated root (prio/add):    32768/00-a0-26-40-0c-e4
Root cost:                     0
Root port:                     Self
Current (root) maximum age:    20 seconds
Current (root) hello time:     2 seconds
Current (root) Forward delay:  15 seconds
Topology change detected:      FALSE
Topology change:               FALSE
```

```
Port      Interface      State
   1      TKR/0          Forwarding
   2      Eth/0          Forwarding
ASRT>
```

· LIST SPANNING-TREE-PROTOCOL TREE

Displays the current spanning tree protocol state information including port, interface and cost information.

Example:

```
ASRT>LIST SPANNING-TREE-PROTOCOL TREE
Port                           Designated  Desig.              Designated  Des.
No. Interface                        Root   Cost                   Bridge  Port
 1  TKR/0         32768/00-a0-26-40-0c-e4      0  32768/00-a0-26-40-0c-e4  80-01
 2  Eth/0         32768/00-a0-26-40-0c-e4      0  32768/00-a0-26-40-0c-e4  80-02
ASRT>
```

i) LIST TRANSPARENT

Displays transparent bridge configuration information. You can display the following general data group options under the **LIST TRANSPARENT** command:

Syntax:

```
ASRT> LIST TRANSPARENT ?
CONFIGURATION
COUNTERS
STATE
```

· LIST TRANSPARENT CONFIGURATION

Displays information concerning the transparent bridge.

Example:

```
ASRT> LIST TRANSPARENT CONFIGURATION
Filtering database size:    2066
Aging time:                 320 seconds
Aging granularity           5 seconds

Port    Interface      State        MTU
   2    Eth/0          Enabled      1514
ASRT>
```

· LIST TRANSPARENT COUNTERS

Displays the transparent bridge counters. Enter **ALL-PORTS** after the command to display the counters for all ports or enter **PORT** and the specific port number after the command to display counters for one port.

Example:

```
ASRT>LIST TRANSPARENT COUNTERS PORT 2
Counters for port 2, interface Eth/0       :
Total frames received by interface:         559984
Frames submitted to bridging:                92964
Frames submitted to routing:                     0
Dropped, source address filtering:               0
Dropped, dest address filtering:            513339
Dropped, protocol filtering:                     0
Dropped, no buffer available to copy:            0
Dropped, input queue overflow:                   0
Dropped, source port blocked:                   84
Frames sent by bridging:                       423
```

```
Dropped, dest port blocked:                      0
Dropped, transmit error:                         0
Dropped, too big to send on port:                0

ASRT>
```

· *LIST TRANSPARENT STATE*

Displays the transparent state information.

Example:

```
ASRT>LIST TRANSPARENT STATE
Filtering database size:          2066
Number of static entries:            2
Number of dynamic entries:         576
Hash collision count:              111
Fitering database overflow:          0
ASRT>
```

2.8. NETBIOS

Displays the NetBIOS monitoring prompt. Enter **NETBIOS** at the ASRT> prompt to display the NetBIOS monitoring prompt.

See Chapter 10 "NetBIOS Filtering and Caching Commands", for an explanation for the NetBIOS commands.

Syntax:

```
ASRT>NETBIOS
```

Example:

```
ASRT>NETBIOS

NetBIOS Support User Console

NetBIOS>
```

Note: If you have not purchased the NetBIOS feature, you receive the following message if you use this command:

```
NetBIOS Support not in load.
```

2.9. NAME-CACHING

Use the **NAME-CACHING** command to enter the Name Caching facility monitoring menu.

Syntax:

```
ASRT>NAME-CACHING
Name Cache>
```

Commands	**Function**
? (HELP)	Displays all the Name Caching monitoring commands, or lists options for specific commands.
LIST	Displays all statistics and counters related to Name Caching.

PORT Selects a specific interface for monitoring purposes.

EXIT Exits the Name Caching monitoring prompt.

a) ? (HELP)

Use the **?** (HELP) command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Example:

```
Name Cache>?
PORT
LIST
EXIT
Name Cache>
```

b) LIST

Use the **LIST** command to display the current Name Caching statistics and counters. This information can be displayed on a global or a per interface basis by using the **PORT** monitoring command.

Syntax:

```
Name Cache>LIST ?
ADD-NAMES
CACHE
```

· LIST ADD-NAMES

Displays the total entries used to filter duplicate Add Name and Add Group Name frames.

Example:

```
Name Cache>LIST ADD-NAMES

                                               Add (Group) Name
        Name                  MAC           Received      Filtered
 -------------------   -----------------   ----------    ----------
 DELL1          <00>   00-00-83-a5-ba-1b            3             2
 NBSDLS         <00>   00-00-83-a5-ba-1b            3             2
 DELL1          <03>   00-00-83-a5-ba-1b            3             2
 DELL1                 00-00-83-a5-ba-1b            3             2
 NBSDLS         <1e>   00-00-83-a5-ba-1b            3             2
 NBSDLS         <1d>   00-00-83-a5-ba-1b            3             2
 ##__MSBROWSE__#<01>   00-00-83-a5-ba-1b            3             2

Name Cache>
```

· LIST CACHE

Syntax:

```
Name Cache>LIST CACHE ?
RIFS
STATISTICS
```

LIST CACHE RIFS

Shows the RIF and MAC information of all known and valid server names.

Example:

```
Name Cache>LIST CACHE RIFS

      Server              MAC Address        Routing Information Field
-------------------    -----------------  -------------------------------
 SOPORTE                Invalid           Invalid
 FYUBERO                Invalid           Invalid


Name Cache>
```

LIST CACHE STATISTICS

Displays the number of times that certain operations have been executed against a particular server name.

Example:

```
Name Cache>LIST CACHE STATISTICS

                                           Broadcasts
       Server            Received    Converted    Forwarded    Filtered
 -------------------    ----------  -----------  -----------  ----------
 SOPORTE                         2            0            2           0
 FYUBERO                         2            0            2           0

Name Cache>
```

c) PORT

Use the **PORT** command to select the bridge’s port to which name caching monitoring commands will be applied.

Example:

```
Name Cache>PORT
Port[1]? 1
Name Cache Port>
```

The following commands are available at the *Name Cache Port>* prompt:

Syntax:

```
Name Cache Port>?
LIST
EXIT
```

· LIST

Syntax:

```
Name Cache Port>LIST ?
ADD-NAMES
CACHE
```

LIST ADD-NAMES

Displays the entries used by a specified port to filter duplicate Add Names and Add Group Names frames.

Example:

```
Name Cache Port>LIST ADD-NAMES

 Add (Group) Name Frames:
    Received         1435
    Filtered         231

Name Cache Port>
```

LIST CACHE

Lists cache counters related to the specified port. These counters are aggregates for all name cache operations on this port.

Example:

```
Name Cache Port>LIST CACHE

 Name Request Broadcast Frames:
    Received         356
    Converted        30
    Forwarded        310
    Filtered         16

Name Cache Port>
```

· *EXIT*

Use the **EXIT** command to return to the Name Cache prompt.

Example:

```
Name Cache Port>EXIT
Name Cache>
```

d) EXIT

Use the **EXIT** command to return to the ASRT prompt.

Example:

```
Name Cache>EXIT
ASRT>
```

2.10. EXIT

Use the **EXIT** command to return to the + prompt.

Syntax:

```
ASRT>EXIT
```

Example:

```
ASRT>EXIT
+
```

Chapter 9
Using NetBIOS

1. About NetBIOS

NetBIOS was designed only for use on a LAN. It is not a routable protocol and is typically bridged or switched using DLSw.

NetBIOS relies on broadcast frames for most of its functions. While this may not present a problem in LAN environments, these broadcasts can be costly in internetwork environments by causing congestion, as well as increased costs for WAN links.

NetBIOS uses LLC type 1 (LLC1) and LLC type 2 (LLC2) services:

- LLC1 provides connectionless data transfer. It requires name conflict resolution, station status gathering flows, and circuit and connection setup flows.
- LLC2 provides a connection-oriented data transfer that uses 1-frame traffic sent on established LLC2 connections.

1.1. NetBIOS Names

NetBIOS names are the key to communication between NetBIOS stations. A NetBIOS station must know its name in order to communicate with other NetBIOS stations.

NetBIOS names have 16 ASCII characters. IBM and Microsoft reserve the 16th character of the NetBIOS name.

There are two types of NetBIOS names:

- Individual names represent a single NetBIOS client or server and should be unique within the NetBIOS network.
- Group names represent a group of NetBIOS stations (an OS/2 LAN Server domain, for example). These names should not be the same as any individual NetBIOS names in the network.

A single NetBIOS station can have multiple individual or group names. The NetBIOS application generates names based on the name or names the network administrator configures.

1.2. NetBIOS Name Conflict Resolution

Before a NetBIOS station uses an individual NetBIOS name, it makes sure that the name is unique. To do so, the station repeatedly broadcasts a Name Conflict Resolution frame to all NetBIOS stations. If the station does not receive a response, it presumes the name is unique and it uses the name.

1.3. NetBIOS Sessions Setup Procedure

To establish a NetBIOS session for data transfer types of operations, the NetBIOS client first determines the MAC address of the NetBIOS server. In Token Ring networks, the client also uses source routing techniques to determine the LLC route to the server.

The following is the process of establishing a session:

1. The client repeatedly broadcasts a Spanning Tree Explorer (STE) NetBIOS UI frame that contains the NetBIOS name of the server to all NetBIOS stations.
2. When the server receives the frame, it responds to the client with a corresponding All Routes Explorer (ARE) NetBIOS UI frame that contains the MAC address of the server and, for Token Ring, the route to the server.

   The client can then do either of the following:

a. Establish an LLC 2 connection to communicate with the server using I-frames.
b. Begin communicating with the server, using specifically-routed NetBIOS UI frames.

2. Reducing NetBIOS Traffic

There are two ways to reduce the amount of broadcast NetBIOS traffic:

- Filter as many broadcast NetBIOS frames as possible.
- Forward unfiltered NetBIOS UI frames on as few bridge ports or DLSw TCP sessions as possible.

The following table lists the NetBIOS filters.

Filter Type	Filters
MAC Address	Frames by either the source or destination MAC address.
Frame Type	Specific types of NetBIOS frames.
Duplicate Frame	Duplicate frames
Response	Responses for which the router did not forward a NetBIOS broadcast frame.
Byte	Frames by byte offset and field length within a frame.
Name	Frames by NetBIOS source and destination names.

Once the router filters frames, name caching and route caching control how the router forwards the remaining frames.

Chapter 12, “Using Mac Filtering,” describes MAC address filtering.

The following sections describe frame type, duplicate frame, and response filtering, name and route caching, and name and byte filtering.

2.1. Frame Type Filtering

Frame type filtering lets you filter the following types of frames:

- Name Conflict Resolution
- General Broadcast
- Trace Control

Name Conflict Resolution

NetBIOS stations use Name Conflict Resolution frames to make sure their name is unique. Name Conflict Resolution frames are Add-Name-Query, Add-Group-Name-Query, Add-Name Response, and Name-In-Conflict.

Use the following guidelines to determine when to filter Name Conflict Resolution frames:

- It is critical that the NetBIOS names of stations to which a NetBIOS session is established (typically a server) be unique.
- It is also usually critical that the individual NetBIOS names of stations within the same group (or domain) be unique.
- It is often not critical that the NetBIOS names of stations from which a NetBIOS session is set up (typically a client) be unique, especially across domains.

  For this reason, networks in which there is good control over server names may gain advantage by filtering name conflict resolution frames. This is especially true for DLSw networks.

General Broadcast

NetBIOS stations use General Broadcast frames to send data to all NetBIOS stations in a network. Stations rarely use this frame, and you can typically filter it. The NetBIOS General Broadcast frame is Datagram-Broadcast.

Trace Control Frames

Trace Control frames terminate NetBIOS traces in all NetBIOS stations in a network. This frame is rarely used and you can typically filter it. The NetBIOS Trace Control frame is Terminate-Trace.

2.2. Configuring Frame Type Filtering

For bridge traffic, the router does not filter any of the above frame types as the default. However, if you are bridging NetBIOS traffic on WAN links, it may be beneficial to filter these frames. To turn frame type filtering on or off for bridging, enter **SET FILTERS BRIDGE**.

For DLSw traffic, the router filters all of the above frame types as the default. To turn frame type filtering on or off for DLSw, enter **SET FILTERS DLSW**.

For example:

```
NetBIOS config>SET FILTERS BRIDGE

Filter Name Conflict frames(Yes/No)(N)? y

Name conflict filtering is              ON

Filter General Broadcast frames(Yes/No)(N)? n

General broadcast filtering is         OFF

Filter Trace Control frames(Yes/No)(N)? y

Trace control filtering is              ON

NetBIOS config>
```

2.3. Duplicate Frame Filtering

When a station sends broadcast frames, it typically sends up to 10 (the default is 6) frames at fixed intervals (the default is 5 seconds).

Duplicate frame filtering causes the router to forward only one instance of each frame within a configurable amount of time. **Figure 9-1** shows how duplicate frame filtering reduces the number of broadcast frames forwarded over the DLSw WAN.

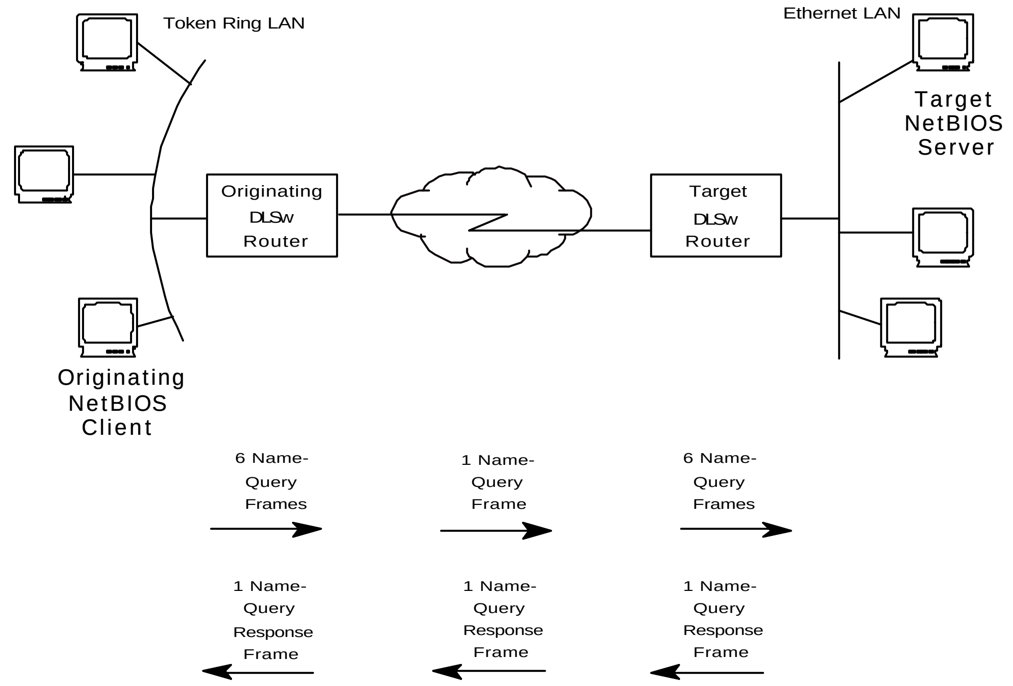


Figure 9-1. Setting Up a NetBIOS Session Over DLSw

Here is the process that the originating NetBIOS client in **Figure 9-1** uses to set up a session with the target NetBIOS server.

1. After verifying that its name is unique, the originating NetBIOS client sends six Name-Query frames at half-second intervals.
2. The originating DLSw router receives the first Name-Query frame and forwards it to the target DLSw router. The originating router filters the remaining five frames.
3. The target DLSw router receives the first Name-Query frame. It then assumes responsibility for setting up the session and sends Name-Query frames to its attached LAN as though it were the originating NetBIOS station.
4. The target NetBIOS station responds to the Name-Query frames with a corresponding Name-Recognized frame that contains its MAC address. For Token Ring frames, the target NetBIOS station also sends the route to the server.
5. The target DLSw router then returns a Specifically-Router Frame (SRF) to the originating DLSw router, which forwards the frame to the originating NetBIOS station.

2.4. How Duplicate Frame Filtering Works

Duplicate frame filtering works by keeping a database of NetBIOS command frame. These include the following: Name-Query, Status-Query, Datagram, Add-Name-Query, Add-Group-Name-Query, and Name-In-Conflict.

Figure 9-2 shows the duplicate frame filtering process for bridge traffic. In this example, the router receives six Name-Query frames in half-second intervals. The Duplicate Frame Filter Timeout is set to 1.5 seconds, and the Duplicate Frame Detect Timeout is set to 5 seconds.

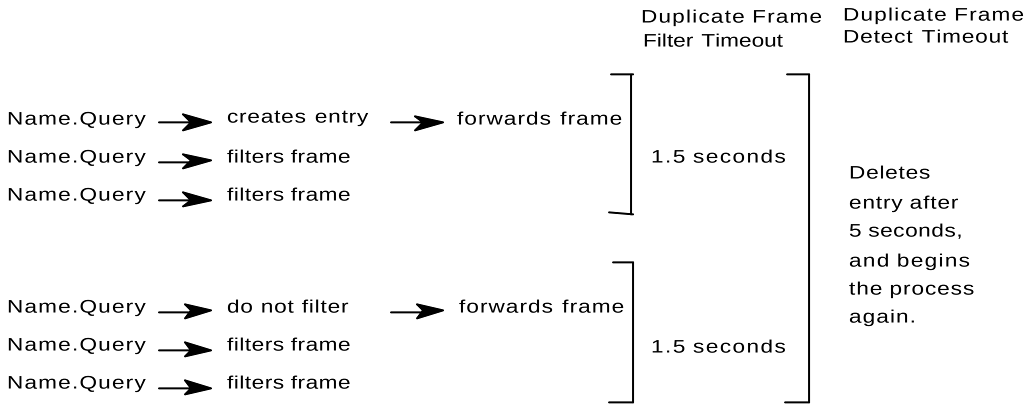


Figure 9-2. Duplicate Frame Filtering Process for Bridged Traffic

Here are the steps for duplicate frame filtering;

1. When the router receives a new frame, it creates an entry for that frame in the duplicate frame database and forwards the frame.
2. The router filters any duplicate frames that it receives within the duplicate frame filter timeout (in this case, 1.5 seconds).
3. If the router receives a duplicate frame after the timer expires, it forwards the frame and resets the timer.

The router repeats this process until the duplicate frame detect timer expires.

For DLSw traffic, the duplicate frame filtering process is the same, except that DLSw does not use the duplicate frame filter timer. DLSw uses only the duplicate frame detect timer. Once the originating router creates an entry, it filters all duplicate frames until the duplicate frame detect timer expires. For DLSw, you can also control how many query frames the target DLSw router sends during a configurable time period.

2.5. Configuring Duplicate Frame Filtering

Duplicate frame filtering is always enabled for DLSw traffic. You cannot enable or disable it.

Duplicate frame filtering is disabled for bridge traffic as the default. You can enable or disable it for bridging using the **ENABLE DUPLICATE-FILTERING** and **DISABLE DUPLICATE-FILTERING** commands.

To change the timers, enter the following:

```
NetBIOS config>SET GENERAL

WARNING! Setting Duplicate Frame Filter Timeout to zero...
         disables duplicate frame checking!

Duplicate frame filter timeout value in seconds[1.5]?
Duplicate frame detect timeout value in seconds[5.0]?

General parameters set

NetBIOS config>
```

WARNING! ***Setting Duplicate Frame Filter Timeout to zero…***

disables duplicate frame checking!

If you enable DLSw, the router also prompts you for the following:

```
Command frame retry count [5]?
Command frame retry timeout value in seconds [0.5]?
```

These parameters control how many query frames the target DLSw router sends during a configured time period.

2.6. Response Frame Filtering

NetBIOS stations expect a response frame to Name-Query and Status-Query frames. If a station does not receive a response, it continues to send queries.

If the router receives a response to a command frame that it did not forward, it drops the response and does not forward it.

You cannot disable response frame filtering on the router.

2.7. Response Frame Filtering for DLSw

For DLSw traffic, make sure the duplicate frame detect timeout is set high enough for the router to have time to set up a session.

As described in section **2.3 "Duplicate Frame Filtering"**, a target DLSw router takes responsibility for setting up a session.

A router takes responsibility for setting up a session if it matches Name-Query and Name-Recognized frames within the duplicate frame detect timeout periods. If the router does not match those frames within that time period, it does not forward the Name-Recognized response frames, and it does not set up the session.

The default duplicate frame detect timeout is five seconds. Do not set the duplicate frame detect timeout to zero, or the router will have no time to set up the session. You can increase the duplicate frame detect timeout using the **SET GENERAL** command.

```
NetBIOS config>SET GENERAL

WARNING! Setting Duplicate Frame Filter Timeout to zero...
         disables duplicate frame checking!

Duplicate frame filter timeout value in seconds[1.5]?
Duplicate frame detect timeout value in seconds[5.0]?

General parameters set

NetBIOS config>
```

WARNING! ***Setting Duplicate Frame Filter Timeout to zero…***

disables duplicate frame checking!

2.8. NetBIOS Name Caching and Route Caching

Name caching and route caching apply to both DLSw and bridging. Once the router filters all possible NetBIOS broadcast frames, it uses NetBIOS name caching and route caching to reduce the number of frames that the router forwards.

With name caching, the router maintains a database of NetBIOS names and routes. Each time the router receives a Name-Recognized frame, it extracts the MAC address and route and enters that information into the database.

When the router receives a Name-Query or Statue-Query, it checks to see if the name being queried is already in its database. If it is, route caching converts the frame from an STE frame to a SRF (Specifically-Routed Frame). A timer on the entry invalidates the database information, if the server does not respond before the timer expires.

2.9. Enabling Caching

Name caching is always enabled. You cannot disable it. The default for route caching is disabled. Enter **ENABLE ROUTE-CACHING** to enable it.

```
NetBIOS config>ENABLE ROUTE-CACHING

Route caching is                        ON

NetBIOS config>
```

2.10. Types of Name Cache Entries

There are three types of name cache entries:

- *Permanent* entries are those that you add at the NetBIOS configuration prompt (`NetBIOS config>`). The router saves permanent entries, and they are still available when you restart the router.
- *Static* entries are those that you enter at the NetBIOS monitoring prompt (`NetBIOS>`). The router does not save static entries, and they are not available after you restart the router.
- *Dynamic* entries are those that the router learns through Name-Query and Name-Recognized processing. A timer removes dynamic entries that are not referenced within a configurable amount of time. The router does not save dynamic entries and they are not available after you restart the router.

There are three types of NetBIOS names kept in the name cache:

- *Individual* is a NetBIOS individual name.
- *Group* is a NetBIOS group name.
- *Unknown* means the router does not yet have information about the name, indicating that a search for the name is not complete.

The router also distinguishes between local and remote entries:

- *Local* is an entry the router can reach locally via the bridge network. The router saves the MAC address associated with the name. If route caching is enabled, the router also saves the best LLC route between the router and the NetBIOS station

- *Remote* is an entry the router can reach remotely via a DLSw TCP session. The router saves the best TCP sessions.

2.11. Adding Name Cache Entries

You can add permanent or static entries for DLSw neighbors to the name caching. Although the router lets you add entries other than DLSw neighbors, it ignores those entries.

You can enter NetBIOS names in ASCII and hexadecimal, either separately or intermixed. For example, you would need to enter an adapter address in hexadecimal mode. The default data entry mode is ASCII. To enter hexadecimal mode, type a left angle bracket (<). To return to ASCII mode, type a right angle bracket (>).

Enter **ADD CACHE-ENTRY** at the `NetBIOS config>` prompt to add static entries.

```
NetBIOS config>ADD CACHE-ENTRY

Enter up to 15 characters of NetBIOS name (no wild cards)[]? nbs<F7>
Enter IP Address [0.0.0.0]? 123.45.67.89

Name cache entry has been created

NetBIOS config>
```

2.12. Setting Cache Parameters

Use the **SET CACHE-PARMS** command to change the following parameters:

```
NetBIOS config>SET CACHE-PARMS

Significant characters in name[16]? 15
Best path aging timeout value in seconds[60.0]?
Reduced search timeout value in seconds[1.5]?
Unreferenced entry timeout value in minutes[5000]?
Max nbr local name cache entries[500]?
Max nbr remote name cache entries[100]?

Cache parameters set

NetBIOS config>
```

See Chapter 10, section 3.8 SET command for more information on the **SET CACHE-PARMS** command.

2.13. Displaying Cache Entries

The router provides the following commands that let you new cache entries.

From the NetBIOS configuration prompt, you can use the **LIST CACHE** commands in **Table 9.1**

Table 9.1 NetBIOS List Cache Configuration Commands

Command	**Displays**
LIST CACHE ALL	All active entries in the router's name cache including permanent, static and dynamic entries.
LIST CACHE ENTRY-NUMBER	A cache entry according to its entry number.
LIST CACHE NAME	A cache entry for a specific NetBIOS name
LIST CACHE IP-ADDRESS	A cache entry for a specific IP address

From the NetBIOS monitoring prompt, you can use the **LIST CACHE** commands in **Table 9.2**

Table 9.2 NetBIOS List Cache Monitoring Commands

Command	**Displays**
LIST CACHE ACTIVE	All active entries in the router's name cache including permanent, static and dynamic entries.
LIST CACHE CONFIG	Static and permanent entries. Does not show dynamic entries.
LIST CACHE GROUP	Entries that exist for NetBIOS group names.
LIST CACHE LOCAL	Local cache entries. Local cache entries are those that the router learns over the bridge.
LIST CACHE NAME	A cache entry for a specific NetBIOS name.
LIST CACHE REMOTE	Remote cache entries. Remote cache entries are those that the router learns over the DLSw WAN.
LIST CACHE UNKNOWN	Entries where the types of NetBIOS entry is unknown

2.14. NetBIOS Name Filtering

NetBIOS name filters apply to both bridging and DLSw. You can use them to filter NetBIOS packets that have specific NetBIOS host names. The router examines the source name or destination name field of the following NetBIOS UI packet types:

- Add-Group-Name-Query (source)
- Add-Name-Query (source)
- Datagram (destination)
- Name-Query (source and destination)

For information on how to create name filters, see **Chapter 11 "Configuring and Monitoring NetBIOS Name and Byte Filters."**

2.15. NetBIOS Byte Filtering

NetBIOS byte filters apply to both bridging and DLSw. Byte filtering lets you filter NetBIOS packets based on fields in the NetBIOS packet.

To build a byte filter, you specify:

- An offset from the beginning of the NetBIOS header
- A byte pattern to match
- An optional mask to apply to the selected fields of the NetBIOS header.

For information on how to create name filters, see **Chapter 11 "Configuring and Monitoring NetBIOS Name and Byte Filters."**

Chapter 10
NetBIOS Filtering and Caching commands

1. About NetBIOS Configuration and Monitoring Commands

Enter NetBIOS configuration commands at the `NetBIOS config>` prompt. This chapter refers to changes you make at the configuration prompt as permanent. However, changes you make at this prompt do not take effect immediately. They become part of the router's configuration memory once you restart the router.

Enter NetBIOS monitoring commands at the `NetBIOS>` prompt. This chapter refers to changes you make at the monitoring prompt as static. Monitoring commands take effect immediately, but the router does not save them after you restart the router.

2. Configuring NetBIOS Filtering and Caching

You can configure the following NetBIOS filtering and caching parameters:

- To configure name caching parameters, enter the **SET CACHE-PARMS** command.
- To configure duplicate frame filtering, enter the **SET GENERAL** command.
- To configure frame type filtering, enter the **SET FILTERS BRIDGE** or **SET FILTERS DLSW** commands.

2.1. Configuring NetBIOS for DLSw

If you are sending NetBIOS traffic over DLSw, you can also configure the following parameter, according to the procedures below:

- Add name cache entries for DLSw neighbors.
- Open NetBIOS SAPs.
- Set a priority for SNA and NetBIOS sessions.
- Set the maximum NetBIOS frame size.
- Set the memory allocation for NetBIOS UI frames.

2.2. Adding Name Cache Entries for DLSw Neighbors

Add name cache entries for DLSw neighbors. You can add multiple entries with different IP addresses for a single NetBIOS name. This allows DLSw to send the frame to multiple DLSw neighbors.

You can enter NetBIOS names in ASCII and hexadecimal, either separately or intermixed. See section **3.3 ADD** command for more information. NetBIOS names are case sensitive and must match the case of the network NetBIOS names.

```
NetBIOS config>ADD CACHE-ENTRY

Enter up to 15 characters of NetBIOS name (no wild cards)[]? accounting<000>
Enter IP Address [0.0.0.0]? 135.77.25.2

Name cache entry has been created

NetBIOS config>
```

2.3. Opening NetBIOS SAPs

At the `DLSw config>` prompt, open NetBIOS SAPs on both sides of the link to enable DLSw to transmit NetBIOS frames.

```
DLSw config>OPEN-SAP
Interface # [0]?
Enter SAP in hex (range 0-F4), 'SNA', 'NB' or 'LNM' [4]?
SAP 4 opened on interface 0
DLSw config>
```

2.4. Setting a Priority for SNA and NetBIOS Sessions

Prioritize SNA and NetBIOS traffic to prevent one type of session from using too much of the available bandwidth during network congestion.

To do so, at the `DLSw config>` prompt enter SET PRIORITY to set a priority of Critical, High, Medium or Low for SNA sessions and NetBIOS sessions. Also, set a message allocation that corresponds to a session's priority.

```
DLSw config>SET PRIORITY
Priority for SNA DLSw sessions (C/H/M/L)[M]? H
Priority for NetBIOS DLSw sessions (C/H/M/L)[M]?
Message allocation by C/H/M/L priority (4 digits)[4/3/2/1]?
Maximum NetBIOS frame size (516, 1470, 2052, or 4399)[2052]?
DLSw config>
```

The router uses the priority and message allocation to selectively limit the burst-length of specific types of traffic. For example, if you assign

- SNA traffic a priority of Critical and Critical sessions have a message allocation of 4, and
- NetBIOS traffic a priority of Medium, and Medium sessions have a message allocation of 2, the router processes 4 SNA frames before it processes 2 NetBIOS frames. Once the router processes 2 NetBIOS frames, it processes 4 SNA frames and so on. In this scenario, the router dedicates two-thirds of available bandwidth to SNA traffic (a ratio of 4 to 2). Note that the router counts frames, rather than bytes, when allocating bandwidth according to the priorities you assign.

You can change the message allocation for sessions from the default of 4/3/2/1. You must always enter four digits, between 1 and 9, in descending order. For example, if the SNA priority is Critical, the NetBIOS traffic is Medium and you change the message allocation to 8/7/6/5, the router processes 8 SNA frames before it processes 6 NetBIOS frames, and so on.

2.5. Setting the Maximum NetBIOS Frame Size

To change the maximum NetBIOS frame size, enter the **SET PRIORITY** command at the `DLSw config>` prompt. The default is 2052. Set this parameter to the largest frame size you expect to needs and no larger. Setting the frame size larger than needed reduces the number of available buffers.

2.6. Setting the Memory Allocation for NetBIOS UI Frames

Enter the **SET MEMORY** command at the `DLSw config>` prompt to set the number of bytes the router allocates as a buffer for NetBIOS UI frames. If the TCP transmit buffer becomes full, the router uses this buffer to collect NetBIOS UI frames.

Note that the number of bytes allocated for NetBIOS is global, and not per session.

```
DLSw config>SET MEMORY
Number of bytes to allocate for DLSw (at least 26624)[141312]?
Number of bytes to allocate per LLC session[8192]?
Number of bytes to allocate per SDLC session[4096]?
Number of bytes to allocate for NetBIOS UI-frames[40960]?
DLSw config>
```

3. NetBIOS Configuration Commands

Table 10.1 lists the NetBIOS configuration commands

Table 10.1 NetBIOS Commands

Command	Function
? (HELP)	Lists available commands or options.
ADD	Adds cache entries to the router’s name cache.
DELETE	Deletes cache entries that you added using the **ADD CACHE-ENTRY** command.
DISABLE	Disables duplicate frame filtering and route caching.
ENABLE	Enables duplicate frame filtering and route caching.
LIST	Displays various cache entries and configuration information.
SET	Configures parameters for name caching, duplicate frame filtering, and frame type filtering. Also, displays the `NETBIOS Filter config>` prompt.
EXIT	Returns to the previous prompt.

3.1. Displaying the NetBIOS Configuration prompt

You can access the `NetBIOS config>` prompt from either the ASRT or DLSw configuration environments.

Changes you make at the `NetBIOS config>` prompt affect both bridging and DLSw.

1. To display the `NetBIOS config>` prompt from the ASRT configuration environment, enter **PROTOCOL ASRT** at the `config>` prompt and **NETBIOS** at the `ASRT config>` prompt.

```
Config>PROTOCOL ASRT

-- ASRT Bridge user configuration --
ASRT config> NETBIOS

NetBIOS Support User Configuration

NetBIOS config>
```

2. To display the `NetBIOS config>` prompt from the DLSw configuration environment, enter **PROTOCOL DLS** at the `config>` prompt and **NETBIOS** at the `DLSw config>` prompt.

```
Config>PROTOCOL DLS
DLSw protocol user configuration
DLSw config>NETBIOS

NetBIOS Support User Configuration

NetBIOS config>
```

3.2. ? (HELP)

Lists available commands or options.

Syntax:

```
NetBIOS config>?
```

Example:

```
NetBIOS config>?
ADD
DELETE
DISABLE
ENABLE
LIST
SET
EXIT
```

3.3. ADD

Adds a new name cache entry to the router’s permanent configuration.

Syntax:

```
NetBIOS config>ADD ?
CACHE-ENTRY
```

a) ADD CACHE-ENTRY

Adds a new entry to the router’s name cache. You can add name cache entries for DLSw neighbors only. The router ignores entries that you add for ASRT traffic.

You can add multiple entries with different IP addresses for a single NetBIOS name. This allows DLSw to send the frame to multiple DLSw neighbors.

You can enter NetBIOS names in ASCII and hexadecimal, either separately or intermixed. For example, you would need to enter an adapter address in hexadecimal mode. The default data entry mode is ASCII. To enter hexadecimal mode, type a left angle bracket (<). To return to ASCII mode, type a right angle bracket (>).

> ***Note: NetBIOS names are case sensitive and must match the case of the network NetBIOS names.***

Example:

```
NetBIOS config>ADD CACHE-ENTRY

Enter up to 15 characters of NetBIOS name (no wild cards)[]? accounting<000>
Enter IP Address [0.0.0.0]? 135.77.25.2

Name cache entry has been created

NetBIOS config>
```

3.4. DELETE

Deletes name cache entries from the router’s permanent configuration. The router prompts for a record number, which is the number of the entry you want to delete. To see a list of entry numbers, enter **LIST CACHE ALL**.

Syntax:

```
NetBIOS config>DELETE CACHE-ENTRY
```

Example:

```
NetBIOS config>DELETE CACHE-ENTRY

Enter name cache record number[1]?

Name cache entry has been deleted

NetBIOS config>
```

3.5. DISABLE

Disables duplicate frame filtering or route caching for the bridge

Syntax:

```
NetBIOS config>DISABLE ?
DUPLICATE-FILTERING
ROUTE-CACHING
```

a) DISABLE DUPLICATE-FILTERING

Disables duplicate frame filtering for bridging. Duplicate frame filtering is always enabled for DLSw traffic. You cannot enable or disable it.

Example:

```
NetBIOS config>DISABLE DUPLICATE-FILTERING

Duplicate frame filtering is           OFF

NetBIOS config>
```

b) DISABLE ROUTE-CACHING

Disables route caching for bridging. Route caching is the process of converting broadcast frames to SRF (Specifically-Routed Frames), using the entries in the NetBIOS name cache. Route caching is always enabled for DLSw traffic. You cannot enable or disable it.

Example:

```
NetBIOS config>DISABLE ROUTE-CACHING

Route caching is                       OFF

NetBIOS config>
```

3.6. ENABLE

Enables duplicate frame filtering or route caching for the bridge.

Syntax:

```
NetBIOS config>ENABLE ?
DUPLICATE-FILTERING
ROUTE-CACHING
```

a) ENABLE DUPLICATE-FILTERING

Enables duplicate frame filtering for bridging. Duplicate frame filtering is always enabled for DLSw traffic. You cannot enable or disable it.

Example:

```
NetBIOS config>ENABLE DUPLICATE-FILTERING

Duplicate frame filtering is            ON

NetBIOS config>
```

b) ENABLE ROUTE-CACHING

Enables route caching for bridging. Route caching is always enabled for DLSw traffic. You cannot enable or disable it. Route caching is the process of converting broadcast frames to Specifically-Routed Frames (SRF), using the entries in the NetBIOS name cache.

Example:

```
NetBIOS config>ENABLE ROUTE-CACHING

Route caching is                        ON

NetBIOS config>
```

3.7. LIST

Displays all the cache entries or displays cache entries by type of entry. Displays filter configuration information or general configuration information.

Syntax:

```
NetBIOS config>LIST ?
CACHE
FILTERS
GENERAL
```

a) LIST CACHE

Syntax:

```
NetBIOS config>LIST CACHE ?
ALL
ENTRY-NUMBER
IP-ADDRESS
NAME
```

· LIST CACHE ALL

Displays all active entries in the router’s permanent name cache. Does not display static or dynamic entries.

The router displays all hexadecimal data in angle brackets. The number in angle brackets just before the IP address is the 16th character of the NetBIOS name. IBM and Microsoft reserve the 16th character of the NetBIOS name, and it always appears in hexadecimal.

Example:

```
NetBIOS config>LIST CACHE ALL

Entry  Name                 IP Address
-----  -------------------  ---------------
    1  test           <00>  1.2.3.4
    2  ejemplo        <00>  145.67.89.10

NetBIOS config>
```

· *LIST CACHE ENTRY-NUMBER*

Displays a cache entry according to its entry number. Enter the **LIST CACHE ALL** command to see a list of all entry numbers.

Example:

```
NetBIOS config>LIST CACHE ENTRY-NUMBER
Enter name cache record number[1]? 2

Entry  Name                 IP Address
-----  -------------------  ---------------
    2  example        <00>  145.67.89.10

NetBIOS config>
```

· *LIST CACHE IP-ADDRESS*

Lets you display an entry for a specific IP address.

Example:

```
NetBIOS config>LIST CACHE IP-ADDRESS
Enter IP Address [0.0.0.0]? 145.67.89.10

Entry  Name                 IP Address
-----  -------------------  ---------------
    2  example        <00>  145.67.89.10

NetBIOS config>
```

· *LIST CACHE NAME*

Displays a cache entry for a specific NetBIOS name. Use the following wildcards to simplify your search:

***** Stands for any character string. For example, "San*" could produce:
San Francisco
Santa Fe
San Juan

? Stands for any one character.

$ Must coincide with the last character in a name.

Following are examples of valid uses of wildcards that match San Francisco:

Fran
San?Fran?isco
*S**

S??*?????????
S'*
S?a?n?F?a?c?s?

**o*	????????????
isco?	Isco$
*San?F**	*

Use as many wildcards as you like, up to the maximum number of characters in a NetBIOS name (15 or 16, depending on how many significant characters you configured using the **SET CACHE-PARMS** command).

> ***Note: NetBIOS are case sensitive.***

Example:

```
NetBIOS config>LIST CACHE NAME
Enter up to 15 characters of NetBIOS name (wild cards ok)[]? test

Entry  Name                 IP Address
-----  -------------------  ---------------
    1  test           <00>  1.2.3.4

NetBIOS config>
```

b) LIST FILTERS

Syntax:

```
NetBIOS config>LIST FILTERS ?
ALL
BRIDGE
DLSW
```

· LIST FILTERS ALL

Displays whether or not frame type filtering is on or off for both bridging and DLSw. Use the **SET FILTERS BRIDGE** and **SET FILTERS DLSW** commands to turn these filters on or off.

Example:

```
NetBIOS config>LIST FILTERS ALL

Bridge name conflict filtering is      OFF
Bridge general bcast filtering is      OFF
Bridge trace control filtering is      OFF

DLS name conflict filtering is          ON
DLS general bcast filtering is          ON
DLS trace control filtering is          ON

NetBIOS config>
```

· LIST FILTERS BRIDGE

Displays whether or not frame type filtering is on or off for bridging. Enter the **SET FILTERS BRIDGE** command to turn these filters on or off.

Example:

```
NetBIOS config>LIST FILTERS BRIDGE

Bridge name conflict filtering is      OFF
Bridge general bcast filtering is      OFF
Bridge trace control filtering is      OFF

NetBIOS config>
```

· LIST FILTERS DLSW

Displays whether or not frame type filtering is on or off for both DLSw. Enter **SET FILTERS DLSW** to turn these filters on or off.

Example:

```
NetBIOS config>LIST FILTERS DLSW

DLS name conflict filtering is          ON
DLS general bcast filtering is          ON
DLS trace control filtering is          ON

NetBIOS config>
```

c) LIST GENERAL

Displays the current NetBIOS caching and filtering configuration.

Syntax:

```
NetBIOS config>LIST GENERAL
```

Example:

```
NetBIOS config>LIST GENERAL

Bridge-only Information:

Bridge duplicate filtering is          OFF
Bridge duplicate frame filter t/o      1.5 seconds

DLS-only Information:

DLS command frame retry count            5
DLS max remote name cache entries      100
DLS command frame retry timeout        0.5 seconds

DLS-Bridge Common Information:

Route caching is                       OFF
Significant characters in name          15
Max local name cache entries           500
Duplicate frame detect timeout         5.0 seconds
Best path aging timeout               60.0 seconds
Reduced search timeout                 1.5 seconds
Unreferenced entry timeout            5000 minutes

NetBIOS config>
```

Note: The DLS-only Information only appears if you enabled DLSw.

3.8. SET

Sets name caching parameters, turns frame type filtering on or off for either bridging or DLSw, and adjusts duplicate frame filtering timers and frame retry timers. Also, displays the NetBIOS name and byte filtering prompt.

Syntax:

```
NetBIOS config>SET ?
CACHE-PARMS
FILTERS
GENERAL
```

a) SET CACHE-PARMS

Sets name caching parameters that apply to bridging or DLSw.

Example:

```
NetBIOS config>SET CACHE-PARMS

Significant characters in name[15]?
Best path aging timeout value in seconds[60.0]?
Reduced search timeout value in seconds[1.5]?
Unreferenced entry timeout value in minutes[5000]?
Max nbr local name cache entries[500]?
Max nbr remote name cache entries[100]?

Cache parameters set

NetBIOS config>
```

Significant characters in name

Determines whether the router considers 15 or 16 characters when it looks up the NetBIOS name. If you enter

- 15, the router ignores the 16th character.
- 16, the router includes the 16th character when it looks up cache entries.

The default is 15.

Best path aging timeout

Amount of time in seconds the router considers the address and route for a local name cache entry to be the best path to that station. When this time expires, the router deletes the name cache entry and attempts to discover a new best path for the NetBIOS name.

To determine the best path, the router considers transmission time between nodes on all possible routes connecting those nodes, as well as largest frame size. The router does not consider a path suitable if it cannot accommodate the largest NetBIOS frame that could be transmitted over the path.

The default is 60 seconds. The range is 1.0 to 100.0 seconds.

Reduced search timeout

When the router receives a Name-Query, Status-Query, or Datagram during the timeout period, it searches based on current NetBIOS name cache information.

If the router receives a duplicate frame after this timer expires, it presumes the previous route is no longer valid and it widens its search. The router forwards the duplicate frame to both bridges and DLSw. DLSw broadcasts the corresponding SSP message to all possible DLSw partners.

The default is 1.5 seconds. The range is 1.0 to 100.0 seconds.

Unreferenced entry timeout

The router keeps a name that is not referenced in its cache for this length of time before deleting it. If the cache fills up, the router removes entries sooner.

The default is 5,000 minutes. The range is 1.0 to 100,000 minutes.

Max nbr local name cache entries

Maximum number of local entries the router saves in the name cache. Local entries are those that the router learns over the bridge.

The default is 500. The range is 1 to 30,000. To optimize memory usage, processor usage, and the amount of broadcast traffic, set this number as close as possible to the total number of NetBIOS stations (servers and clients) that are active on this router's local bridge network.

Max nbr remote name cache entries

Maximum number of remotely-learned entries, group name entries and unknown entries.

The default is 100. The range is 1 to 30,000. To optimize memory usage, processor usage, and the amount of broadcast traffic, set this number to the number of remote NetBIOS clients on this router’s local bridge network, plus about 25%.

b) SET FILTERS

Syntax:

```
NetBIOS config>SET FILTERS ?
BRIDGE
BYTE
DLSW
NAME
```

· SET FILTERS BRIDGE

Turns on or off frame-type filtering for bridging

Example:

```
NetBIOS config>SET FILTERS BRIDGE

Filter Name Conflict frames(Yes/No)(N)?

Name conflict filtering is             OFF

Filter General Broadcast frames(Yes/No)(N)? y

General broadcast filtering is          ON

Filter Trace Control frames(Yes/No)(N)?

Trace control filtering is             OFF

NetBIOS config>
```

· SET FILTERS BYTE

From the `NetBIOS config>` prompt, displays the NetBIOS filtering configuration prompt (`NETBIOS Filter config>`).

This prompt allows you to set up NetBIOS byte filters.

See **Chapter 11 “Configuring and Monitoring NetBIOS Name and Byte Filters,”** for more information on the commands available at this prompt.

Example:

```
NetBIOS config>SET FILTERS BYTE
NETBIOS Filtering configuration
NETBIOS Filter config>
```

· SET FILTERS DLSW

Sets frame-type filters for DLSw traffic.

Example:

```
NetBIOS config>SET FILTERS DLSW

Filter Name Conflict frames(Yes/No)(Y)? y

Name conflict filtering is              ON

Filter General Broadcast frames(Yes/No)(Y)? n

```

```
General broadcast filtering is         OFF

Filter Trace Control frames(Yes/No)(Y)? n

Trace control filtering is             OFF

NetBIOS config>
```

· SET FILTERS NAME

From the `NetBIOS config>` prompt, displays the `NETBIOS Filter config>` prompt. This prompt allows you to set up NetBIOS name filters.

See **Chapter 11 “Configuring and Monitoring NetBIOS Name and Byte Filters,”** for more information on the commands available at this prompt.

Example:

```
NetBIOS config>SET FILTERS NAME
NETBIOS Filtering configuration
NETBIOS Filter config>
```

c) SET GENERAL

Sets the duplicate frame timeout, duplicate frame detect timeout, and the command frame retry count and timeout. See **Section 2.3 “Duplicate Frame Filtering”** on Chapter 9 for more information on how duplicate frame filters work.

Example:

```
NetBIOS config>SET GENERAL

WARNING! Setting Duplicate Frame Filter Timeout to zero...
         disables duplicate frame checking!

Duplicate frame filter timeout value in seconds[1.5]?
Duplicate frame detect timeout value in seconds[5.0]?
Command frame retry count[5]?
Command frame retry timeout value in seconds[0.5]?

General parameters set

NetBIOS config>
```

WARNING! ***Setting Duplicate Frame Filter Timeout to zero…***
disables duplicate frame checking!

If DLSw is **not** enabled, the software does **not** display the following:

```
Command frame retry count[5]?
Command frame retry timeout value in seconds[0.5]?
```

Duplicate frame filter timeout

Applies only to bridged traffic if duplicate-filtering is enabled.

During this timeout period, the router filters all duplicate frames it receives.

The range is 0.0 to 100.000 seconds. Zero disables duplicate frame checking. The default is 1.5 seconds.

Duplicate frame detect timeout

Applies to both bridged and DLSw traffic.

Amount of time the router saves entries in its duplicate frame filter database. When this timer expires, the router creates new entries for new frames that it receives.

The range is 0.0 to 100.000 seconds. The default is 5 seconds.

Command frame retry count

Applies to DLSw traffic.

Number of duplicate NetBIOS UI frames the target DLSw router sends to its locally-attached LAN. The router sends these frames at intervals specified by the *command frame retry timeout.*

The range is 0.0 to 10. The default is 5 seconds.

Command frame retry timeout

Applies to DLSw traffic.

This is the interval at which a neighbor DLSw router retries sending duplicate NetBIOS UI frames to its local bridge network.

The range is 0.0 to 10.00 seconds. The default is 5 seconds.

3.9. EXIT

Returns to the previous prompt.

Syntax:

```
NetBIOS config>EXIT
```

Example:

```
NetBIOS config>EXIT
ASRT config>
```

4. NetBIOS Monitoring Commands

Table 10.2 lists the NetBIOS Monitoring commands

Table 10.2. NetBIOS Monitoring Commands

Command	Function
? (HELP)	Lists available commands or options.
ADD	Adds cache entries to the router’s name cache.
DELETE	Deletes cache entries that you added using the **ADD CACHE-ENTRY** command.
DISABLE	Disables duplicate frame filtering and route caching.
ENABLE	Enables duplicate frame filtering and route caching.
LIST	Displays various cache entries and monitoring information.
SET	Configures parameters for name caching, duplicate frame filtering, and frame type filtering. Also, displays the `NETBIOS Filter>` prompt.
EXIT	Returns to the previous prompt.

4.1. Displaying the NetBIOS Monitoring prompt

You can access the `NetBIOS>` prompt from either the ASRT or DLSw monitoring environments. Changes you make at the NetBIOS> prompt affect both bridging and DLSw.

1. To display the `NetBIOS >` prompt from the ASRT monitoring environment, enter **PROTOCOL ASRT** at the + prompt and **NETBIOS** at the `ASRT>` prompt.

```
+PROTOCOL ASRT
ASRT>NETBIOS

NetBIOS Support User Console

NetBIOS>
```

2. To display the `NetBIOS>` prompt from the DLSw monitoring environment, enter **PROTOCOL DLS** at the + prompt and **NETBIOS** at the `DLSw>` prompt.

```
+PROTOCOL DLS
DLSw>NETBIOS

NetBIOS Support User Console

NetBIOS>
```

4.2. ? (HELP)

Lists available commands or options.

Syntax:

```
NetBIOS>?
```

Example:

```
NetBIOS>?
ADD
DELETE
DISABLE
ENABLE
LIST
SET
EXIT
```

4.3. ADD

Adds a new name cache entry to the router's static configuration.

Syntax:

```
NetBIOS>ADD ?
CACHE-ENTRY
```

a) ADD CACHE-ENTRY

Adds a new entry to the router's name cache. You can add name cache entries for DLSw neighbors only. The router ignores entries that you add for ASRT traffic.

You can add multiple entries with different IP addresses for a single NetBIOS name. This allows DLSw to send the frame to multiple DLSw neighbors.

You can enter NetBIOS names in ASCII and hexadecimal, either separately or intermixed. For example, you would need to enter an adapter address in hexadecimal mode. The default data entry mode is ASCII. To enter hexadecimal mode, type a left angle bracket (<). To return to ASCII mode, type a right angle bracket (>).

Note: NetBIOS names are case sensitive and must match the case of the network NetBIOS names.

Example:

```
NetBIOS>ADD CACHE-ENTRY

Enter up to 15 characters of NetBIOS name (no wild cards)[]? accounting<000>
Enter IP Address [0.0.0.0]? 135.77.25.2

Name cache entry has been created

NetBIOS config>
```

4.4. DELETE

Deletes name cache entries from the router's static configuration or active cache. The router prompts for a cache entry name. To see a list of entries, enter **LIST CACHE CONF** or **LIST CACHE ACTIVE**.

Note: NetBIOS names are case sensitive.

Syntax:

```
NetBIOS>DELETE CACHE-ENTRY
```

Example:

```
NetBIOS>DELETE CACHE-ENTRY

Enter up to 15 characters of NetBIOS name (no wild cards)[]? test

Name cache entry NOT found in Active list for name entered
Name cache entry has NOT been deleted from Active list

Static name cache entry deleted from temporary config list

NetBIOS>
```

4.5. DISABLE

Disables duplicate frame filtering or route caching for the bridge

Syntax:

```
NetBIOS>DISABLE ?
DUPLICATE-FILTERING
ROUTE-CACHING
```

a) *DISABLE DUPLICATE-FILTERING*

Disables duplicate frame filtering for bridging. Duplicate frame filtering is always enabled for DLSw traffic. You cannot enable or disable it.

Example:

```
NetBIOS>DISABLE DUPLICATE-FILTERING

Duplicate frame filtering is           OFF

NetBIOS>
```

b) *DISABLE ROUTE-CACHING*

Disables route caching for bridging. Route caching is the process of converting broadcast frames to Specifically-Routed Frames (SRF), using the entries in the NetBIOS name cache. Route caching is always enabled for DLSw traffic. You cannot enable or disable it.

Example:

```
NetBIOS>DISABLE ROUTE-CACHING

Route caching is                       OFF

NetBIOS>
```

4.6. ENABLE

Enables duplicate frame filtering or route caching for the bridge.

Syntax:

```
NetBIOS>ENABLE ?
DUPLICATE-FILTERING
ROUTE-CACHING
```

a) ENABLE DUPLICATE-FILTERING

Enables duplicate frame filtering for bridging. Duplicate frame filtering is always enabled for DLSw traffic. You cannot enable or disable it.

Example:

```
NetBIOS>ENABLE DUPLICATE-FILTERING

Duplicate frame filtering is            ON

NetBIOS>
```

b) ENABLE ROUTE-CACHING

Enables route caching for bridging. Route caching is always enabled for DLSw traffic. You cannot enable or disable it. Route caching is the process of converting broadcast frames to Specifically-Routed Frames (SRF), using the entries in the NetBIOS name cache.

Example:

```
NetBIOS>ENABLE ROUTE-CACHING

Route caching is                        ON

NetBIOS>
```

4.7. LIST

Displays various types of cache entries, filter configuration, general monitoring information, or statistics on caching and filtering.

Syntax:

```
NetBIOS>LIST ?
CACHE
FILTERS
GENERAL
STATISTICS
```

a) LIST CACHE

Syntax:

```
NetBIOS>LIST CACHE ?
ACTIVE
CONFIG
GROUP
LOCAL
NAME
REMOTE
UNKNOWN
```

· LIST CACHE ACTIVE

Displays all active entries in the router’s name cache, including dynamic, static and permanent entries.

The router displays all hexadecimal data in angle brackets. The number in angle brackets just before the IP address is the 16th character of the NetBIOS name. IBM and Microsoft reserve the 16th character of the NetBIOS name, and it always appears in hexadecimal.

If the Name Type field does not specify local, it is a remote entry. For a description of the fields in this display, see the **LIST CACHE NAME** command on this section.

Example:

```
NetBIOS>LIST CACHE ACTIVE

Cnt  NetBIOS Name         Name Type          Entry Type
---  -------------------  -----------------  ----------
  1  ADMIN          <00>  INDIVIDUAL LOCAL   DYNAMIC
  2  MAILER         <20>  UNKNOWN            DYNAMIC
  3  DEV            <1b>  UNKNOWN            DYNAMIC
  4  RESEARCH       <1b>  UNKNOWN            DYNAMIC
  5  JOHN           <00>  INDIVIDUAL LOCAL   DYNAMIC
  6  JAXE           <00>  INDIVIDUAL LOCAL   DYNAMIC
  7  LABNT          <00>  INDIVIDUAL LOCAL   DYNAMIC

NetBIOS>
```

· *LIST CACHE CONFIG*

Displays all static and permanent name cache entries. Does not show dynamic entries.

The router displays all hexadecimal data in angle brackets. The number in angle brackets just before the IP address is the 16th character of the NetBIOS name. IBM and Microsoft reserve the 16th character of the NetBIOS name, and it always appears in hexadecimal.

Example:

```
NetBIOS>LIST CACHE CONFIG

Name                 IP Address       Source     Last Mod
-------------------  ---------------  ---------  ---------
SHEPHERD       <00>  192.9.1.134  PERMANENT  UNCHANGED

NetBIOS>
```

· *LIST CACHE GROUP*

Displays cache entries that exist for NetBIOS group names. For a description of the fields in this display, see the **LIST CACHE NAME** command on this section.

Example:

```
NetBIOS>LIST CACHE GROUP

Cnt  NetBIOS Name         Entry Type  Rem Path St  IP Address(es)
---  -------------------  ----------  -----------  ---------------
  1  ID             <1d>  DYNAMIC     GROUP

NetBIOS>
```

· *LIST CACHE LOCAL*

Displays the local cache entries. Local cache entries are those that the router learns via the local bridge network. For a description of the fields in this display, see the **LIST CACHE NAME** command on this section.

For NetBIOS clients the Local Path State is always Unknown and the MAC Address and Routing Information fields are always empty.

Example:

```
NetBIOS>LIST CACHE LOCAL

Cnt  NetBIOS Name         Loc Path St  MAC Address   Routing Information
---  -------------------  -----------  ------------  ------------------------
  1  MARTINS        <00>  UNKNOWN
  2  LAB486         <00>  UNKNOWN
  3  MABERED        <20>  UNKNOWN
  4  TEL0106        <20>  UNKNOWN
  5  TSERVER        <06>  UNKNOWN

NetBIOS>
```

· LIST CACHE NAME

Displays a cache entry for a specific NetBIOS name. Use the following wildcards to simplify your search:

***** Stands for any character string. For example, “San*” could produce:
San Francisco
Santa Fe
San Juan

? Stands for any one character.

$ Must coincide with the last character in a name.

Following are examples of valid uses of wildcards that match San Francisco:

Fran
San?Fran?isco
*S**
**o*
isco?
*San?F**

S??*?????????
S’*
S?a?n?F?a?c?s?
????????????
Isco$
*

Use as many wildcards as you like, up to the maximum number of characters in a NetBIOS name (15 or 16, depending on how many significant characters you configured using the **SET CACHE-PARMS** command).

> ***Note: NetBIOS are case sensitive.***

Example:

```
NetBIOS>LIST CACHE NAME
Enter up to 15 characters of NetBIOS name (wild cards ok)[]? TEST

NetBIOS Name         Name Type          Entry Type
-------------------  -----------------  ----------
TEST           <00>  INDIVIDUAL LOCAL   DYNAMIC

    Count of name cache entry hits:     0

    Age of name cache entry:            137535
    Age of name cache last reference:   137536

    Local path information:

        Loc Path St  Timestamp  MAC Address    LFS   Routing Information
        -----------  ---------  ------------  -----  -------------------
        UNKNOWN      254372
    Remote path information:

        Rem Path St  Timestamp   LFS   IP Address(es)
        -----------  ---------  -----  -------------
        UNKNOWN      254374

Do you wish to continue(Yes/No)(Y)? y
NetBIOS>
```

NetBIOS Name The entry's NetBIOS name.

Name Type Type of NetBIOS name. Possible types are

INDIVIDUAL NetBIOS individual name.

GROUP NetBIOS group name.

UNKNOWN The router does not have information about the name, indicating that a search for the name is not complete.

LOCAL An entry the router can reach locally via the bridge network.

REMOTE An entry the router can reach remotely via a DLSW TCP session.

Entry Type Possible entry types are

PERMANENT Entries that you add at the `NetBIOS config>` prompt using the **ADD CACHE-ENTRY** command.

STATIC Entries that you add at the `NetBIOS>` prompt using the **ADD CACHE ENTRY** command.

DYNAMIC Entries that the router learns through Name-Query and Name-Recognized processing.

Count of name cache entry hits Number of times the entry was referenced.

Age of name cache entry Number of timer ticks since the entry was added. Timer ticks vary according to hardware platform.

Age of name cache last Reference Number of timer ticks since an entry was added. Timer ticks vary according to the hardware platform.

Local path information:

Loc Path St Local Path State. Best possible states are

BEST FOUND The router found the best route to this station.

UNKNOWN The router has not yet found the best route to this station.

GROUP The router does not search for a best path for group names.

SEARCH LTD The router is conducting a limited search for this NetBIOS name. See the **SET CACHE-PARMS** command for more information on a reduced search.

SEARCH ALL The router is conducting a full search. When the **SET CACHE-PARMS** command's reduced search timer expires, the router conducts a full search.

Timestamp Number of timer ticks since the software last updated an entry. Timer ticks vary according to hardware platform.

MAC Address If the entry is a server, displays the MAC address of the server.

LSF Largest Frame Size that the router can use for the entry.

Routing Information Displays standard Routing Information Field (RIF) information.

Remote Path Information

Rem Path St Remote Path State. Possible states are the following

BEST FOUND The router found the best route to this station.

UNKNOWN The router has not yet found the best route to this station.

GROUP The router does not search for a best path for group names.

SEARCH LTD The router is conducting a limited search for this NetBIOS name. See the **SET CACHE-PARMS** command for more information on a reduced search.

SEARCH ALL The router is conducting a full search. When the **SET CACHE-PARMS** command's reduced search timer expires, the router conducts a full search.

Timestamp Number of timer ticks since an entry was last updated. Timer ticks vary according to hardware platform.

LSF Largest Frame Size that the router can use for the entry.

IP Address IP address of the DLSw partner.

· *LIST CACHE REMOTE*

Displays cache entries the router learns over the DLSw WAN. If the router has found the best path, it displays the IP address associated with the DLSw neighbor that can reach the NetBIOS station. For a description of the fields in this display, see the **LIST CACHE NAME** command on this section.

Example:

```
NetBIOS>LIST CACHE REMOTE

Cnt  NetBIOS Name         Entry Type  Rem Path St  IP Address(es)
---  -------------------  ----------  -----------  ---------------
  1  FIRMWARE       <1e>  DYNAMIC     BEST FOUND   20.55.27.33

NetBIOS>
```

· *LIST CACHE UNKNOWN*

Displays cache entries where the type of NetBIOS name is unknown. The router enters all dynamic entries as Unknown until it learns the type of name. It then marks entries as local, remote, or group. For a description of the fields in this display, see the **LIST CACHE NAME** command on this section.

Example:

```
NetBIOS>LIST CACHE UNKNOWN

Cnt  NetBIOS Name         Entry Type  Loc Path St  Rem Path St  IP Address(es)
---  -------------------  ----------  -----------  -----------  --------------
  1  CBRA           <1d>  DYNAMIC     UNKNOWN      SEARCH ALL
  2  HARDWARE       <1e>  DYNAMIC     UNKNOWN      SEARCH ALL
  3  JSPNRMPTGSBSSDI<52>  DYNAMIC     UNKNOWN      SEARCH ALL
  4  TEL01          <00>  DYNAMIC     UNKNOWN      SEARCH LTD

NetBIOS>
```

b) *<u>LIST FILTERS</u>*

Syntaxis:

```
NetBIOS>LIST FILTERS ?
ALL
BRIDGE
DLSW
```

· *LIST FILTERS ALL*

Displays whether or not frame type filtering is on or off for both bridging and DLSw. Use the **SET FILTERS BRIDGE** and **SET FILTERS DLSW** commands to turn these filters on or off.

Example:

```
NetBIOS>LIST FILTERS ALL

Bridge name conflict filtering is      OFF
Bridge general bcast filtering is      OFF
Bridge trace control filtering is      OFF

DLS name conflict filtering is          ON
DLS general bcast filtering is          ON
DLS trace control filtering is          ON

NetBIOS>
```

· *LIST FILTERS BRIDGE*

Displays whether or not frame type filtering is on or off for bridging. Use the **SET FILTERS BRIDGE** command to turn these filters on or off.

Example:

```
NetBIOS>LIST FILTERS BRIDGE

Bridge name conflict filtering is      OFF
Bridge general bcast filtering is      OFF
Bridge trace control filtering is      OFF

NetBIOS>
```

· *LIST FILTERS DLSW*

Displays whether or not frame type filtering is on or off for both DLSw. Use the **SET FILTERS DLSW** command to turn these filters on or off.

Example:

```
NetBIOS>LIST FILTERS DLSW

DLS name conflict filtering is          ON
DLS general bcast filtering is          ON
DLS trace control filtering is          ON

NetBIOS>
```

c) LIST GENERAL

Displays the current NetBIOS caching and filtering monitoring.

Example:

```
NetBIOS>LIST GENERAL

Bridge-only Information:

Bridge duplicate filtering is          OFF
Bridge duplicate frame filter t/o      1.5 seconds

DLS-only Information:

DLS command frame retry count            5
DLS max remote name cache entries      100
DLS command frame retry timeout        0.5 seconds

DLS-Bridge Common Information:

Route caching is                       OFF
Significant characters in name          15
Max local name cache entries           500
Duplicate frame detect timeout         5.0 seconds
Best path aging timeout               60.0 seconds
Reduced search timeout                 1.5 seconds
Unreferenced entry timeout            5000 minutes

NetBIOS>
```

> ***Note: The DLS-only Information only appears if you enabled DLSw.***

d) LIST STATISTICS

Syntax:

```
NetBIOS>LIST STATISTICS ?
CACHE
FRAMES
GENERAL
```

· LIST STATISTICS CACHE

Lists name cache statistics.

Example:

```
NetBIOS>LIST STATISTICS CACHE

Local name cache entries                 2
Remote name cache entries                1
Local individual names                   1
Remote individual names                  0
Group names                              0
Unknown names                            1
Name cache hits                          2312
Name cache misses                        3

NetBIOS>
```

· LIST STATISTICS FRAMES

Syntax:

```
NetBIOS>LIST STATISTICS FRAMES ?
BRIDGE
DLSW
```

LIST STATISTICS FRAMES BRIDGE

Lists name cache statistics for bridging.

Example:

```
NetBIOS>LIST STATISTICS FRAMES BRIDGE

Frames in cache                         3
Name query frames                       2
Status query frames                     1
Add name frames                         0
Add group name frames                   0
Name in conflict frames                 0
Frames not filtered as duplicates       0

NetBIOS>
```

LIST STATISTICS FRAMES DLSW

Lists name cache statistics for DLSw.

Example:

```
NetBIOS>LIST STATISTICS FRAMES DLSW

Name query frames                       0
Status query frames                     0
Add name frames                         0
Add group name frames                   0
Name in conflict frames                 0
Frames not filtered as duplicates       0

NetBIOS>
```

· *LIST STATISTICS GENERAL*

Syntax:

```
NetBIOS>LIST STATISTICS GENERAL ?
BRIDGE
DLSW
```

LIST STATISTICS GENERAL BRIDGE

Displays frame counts for bridging.

Example:

```
NetBIOS>LIST STATISTICS GENERAL BRIDGE

Frames received                     46705
Frames discarded                        0
Frames forwarded to bridge          46705
Frames forwarded to DLS             43716

NetBIOS>
```

LIST STATISTICS GENERAL DLSW

Displays frame counts for DLSw.

Example:

```
NetBIOS>LIST STATISTICS GENERAL DLSW

Frames received                          0
Frames discarded                         0
Frames forwarded to bridge               0

NetBIOS>
```

4.8. SET

Sets name caching parameters, turns frame type filtering on or off for either bridging or DLSw, and adjusts duplicate frame filtering timers and frame retry timers. Also, displays the NetBIOS name and byte filtering prompt.

Syntax:

```
NetBIOS>SET ?
CACHE-PARMS
FILTERS
GENERAL
```

a) SET CACHE-PARMS

Sets name caching parameters that apply to bridging or DLSw.

Example:

```
NetBIOS>SET CACHE-PARMS

Significant characters in name[15]?
Best path aging timeout value in seconds[60.0]?
Reduced search timeout value in seconds[1.5]?
Unreferenced entry timeout value in minutes[5000]?
Max nbr local name cache entries[500]?
Max nbr remote name cache entries[100]?

Cache parameters set

NetBIOS>
```

Significant characters in name

Determines whether the router considers 15 or 16 characters when it looks up the NetBIOS name. If you enter.

- 15, the router ignores the 16th character.
- 16, the router includes the 16th character when it looks up cache entries.

The default is 15.

Best path aging timeout

Amount of time in seconds the router considers the address and route for a local name cache entry to be the best path to that station. When this time expires, the router deletes the name cache entry and attempts to discover a new best path for the NetBIOS name.

To determine the best path, the router considers transmission time between nodes on all possible routes connecting those nodes, as well as largest frame size. The router does not consider a path suitable if it cannot accommodate the largest NetBIOS frame that could be transmitted over the path.

The default is 60 seconds. The range is 1.0 to 100.0 seconds.

Reduced search timeout

When the router receives a Name-Query, Status-Query, or Datagram during the timeout period, it searches based on current NetBIOS name cache information.

If the router receives a duplicate frame after this timer expires, it presumes the previous route is no longer valid and it widens its search. The router forwards the duplicate frame to both bridges and DLSw. DLSw broadcasts the corresponding SSP message to all possible DLSw partners.

The default is 1.5 seconds. The range is 1.0 to 100.0 seconds.

Unreferenced entry timeout The router keeps a name that is not referenced in its cache for this length of time before deleting it. If the cache fills up, the router removes entries sooner.

The default is 5,000 minutes. The range is 1.0 to 100,000 minutes.

Max nbr local name cache entries Maximum number of local entries the router saves in the name cache. Local entries are those that the router learns over the bridge.

The default is 500. The range is 1 to 30,000. To optimize memory usage, processor usage, and the amount of broadcast traffic, set this number as close as possible to the total number of NetBIOS stations (servers and clients) that are active on this router's local bridge network.

Max nbr remote name cache entries Maximum number of remotely-learned entries, group name entries and unknown entries.

The default is 100. The range is 1 to 30,000. To optimize memory usage, processor usage, and the amount of broadcast traffic, set this number to the number of remote NetBIOS clients on this router's local bridge network, plus about 25%.

b) SET FILTERS

Syntax:

```
NetBIOS>SET FILTERS ?
BRIDGE
BYTE
DLSW
NAME
```

· SET FILTERS BRIDGE

Turns on or off frame-type filtering for bridging

Example:

```
NetBIOS>SET FILTERS BRIDGE

Filter Name Conflict frames(Yes/No)(N)?

Name conflict filtering is             OFF

Filter General Broadcast frames(Yes/No)(N)? y

General broadcast filtering is          ON

Filter Trace Control frames(Yes/No)(N)?

Trace control filtering is             OFF

NetBIOS>
```

· SET FILTERS BYTE

From the `NetBIOS>` prompt, displays the NetBIOS filtering monitoring prompt (`NETBIOS Filter>`).

This prompt allows you to set up NetBIOS byte filters.

See **Chapter 11 "Configuring and Monitoring NetBIOS Name and Byte Filters,"** for more information on the commands available at this prompt.

Example:

```
NetBIOS>SET FILTERS BYTE
NETBIOS Filtering configuration
NETBIOS Filter>
```

· *SET FILTERS DLSW*

Sets frame-type filters for DLSw traffic.

Example:

```
NetBIOS>SET FILTERS DLSW

Filter Name Conflict frames(Yes/No)(Y)? y

Name conflict filtering is              ON

Filter General Broadcast frames(Yes/No)(Y)? n

General broadcast filtering is         OFF

Filter Trace Control frames(Yes/No)(Y)? n

Trace control filtering is             OFF

NetBIOS>
```

· *SET FILTERS NAME*

From the `NetBIOS>` prompt, displays the `NETBIOS Filter>` prompt.

This prompt allows you to set up NetBIOS name filters.

See **Chapter 11 “Configuring and Monitoring NetBIOS Name and Byte Filters,”** for more information on the commands available at this prompt.

Example:

```
NetBIOS>SET FILTERS NAME
NETBIOS Filtering configuration
NETBIOS Filter>
```

c) SET GENERAL

Sets the duplicate frame timeout, duplicate frame detect timeout, and the command frame retry count and timeout. See section **2.3 “Duplicate Frame Filtering”** on Chapter 9 for more information on how duplicate frame filters work.

Example:

```
NetBIOS>SET GENERAL

WARNING! Setting Duplicate Frame Filter Timeout to zero...
         disables duplicate frame checking!

Duplicate frame filter timeout value in seconds[1.5]?
Duplicate frame detect timeout value in seconds[5.0]?
Command frame retry count[5]?
Command frame retry timeout value in seconds[0.5]?

General parameters set

NetBIOS>
```

WARNING! ***Setting Duplicate Frame Filter Timeout to zero…***

disables duplicate frame checking!

If DLSw is **not** enabled, the software does **not** display the following:

```
Command frame retry count[5]?
Command frame retry timeout value in seconds[0.5]?
```

Duplicate frame filter timeout

Applies only to bridged traffic if duplicate-filtering is enabled.

During this timeout period, the router filters all duplicate frames it receives.

The range is 0.0 to 100.000 seconds. Zero disables duplicate frame checking. The default is 1.5 seconds.

Duplicate frame detect timeout

Applies to both bridged and DLSw traffic.

Amount of time the router saves entries in its duplicate frame filter database. When this timer expires, the router creates new entries for new frames that it receives.

The range is 0.0 to 100.000 seconds. The default is 5 seconds.

Command frame retry count

Applies to DLSw traffic.

Number of duplicate NetBIOS UI frames the target DLSw router sends to its locally-attached LAN. The router sends these frames at intervals specified by the *command frame retry timeout.*

The range is 0.0 to 10. The default is 5 seconds.

Command frame retry timeout

Applies to DLSw traffic.

This is the interval at which a neighbor DLSw router retries sending duplicate NetBIOS UI frames to its local bridge network.

The range is 0.0 to 10.00 seconds. The default is 5 seconds.

4.9. EXIT

Returns to the previous prompt.

Syntax:

```
NetBIOS>EXIT
```

Example:

```
NetBIOS>EXIT
ASRT>
```

Chapter 11
Configuration and Monitoring NetBIOS Name and Byte Filters

1. Displaying the NetBIOS Filtering Prompts

This section describes the NetBIOs Name and Byte filter configuration and monitoring commands.

Enter configuration commands at the `NetBIOS Filter config>` prompt. Display this prompt as follows:

```
Config>PROTOCOL ASRT

-- ASRT Bridge user configuration --
ASRT config>NETBIOS

NetBIOS Support User Configuration

NetBIOS config>SET FILTERS NAME
NETBIOS Filtering configuration
NETBIOS Filter config>
```

Enter monitoring commands at the `NetBIOS Filter>` prompt. Display this prompt as follows:

```
ASRT>NETBIOS

NetBIOS Support User Console

NetBIOS>SET FILTERS NAME
NETBIOS Filter>
```

2. Setting Up NetBIOS Name and Byte Filters

A name or byte filter is made up of

- Filter lists, which are made up of one or more filter items
- Filter items, which specify the NetBIOS names you want to filter

The router compares each filter item against a packet in the order in which you enter the filter items.

You configure the NetBIOS name and byte filters for each port and specify whether the filter applies to input or output packets.

The following sections provide examples of how to set up a host name filter and a byte filter. The "**NetBIOS Name and Byte Filter Configuration Commands**" and "**NetBIOS Name and Byte Filter Monitoring Commands**" sections describe the commands used in these examples.

Example 1:

Use the following procedure as a guideline to create a name filter. Before you begin, display the `NETBIOS Filter config>` prompt.

```
Config>PROTOCOL ASRT

-- ASRT Bridge user configuration --
ASRT config>NETBIOS

NetBIOS Support User Configuration

NetBIOS config>SET FILTERS NAME
NETBIOS Filtering configuration
NETBIOS Filter config>
```

1. Create an empty name filter list.

   Enter **CREATE NAME-FILTER-LIST**. The software prompts you to name your filter list.

```
NETBIOS Filter config>CREATE NAME-FILTER-LIST
Handle for Name Filter List []?boston
```

2. Display the configuration prompt for the filter list you just created. Enter **UPDATE**. The router prompts you for the name of the filter list.

```
NETBIOS Filter config>UPDATE
Handle for Filter List []?boston
Name Filter List Configuration
NETBIOS Name boston config>
```

3. Add filter items to the filter list.

   When you add a filter item, you must specify the following parameters in this order:

   - *Inclusive* (bridge) or *exclusive* (dropped).
   - ASCII or *hex* is how you enter the name.
   - *Hostname* is the actual name in either an ASCII or hex format. This entry is case sensitive.
   - *Special 16th character* is an optional parameter for use with ASCII strings containing fewer than 16 characters.

The following example adds a filter item to the filter list boston, which allows packets containing the name westboro (an ASCII string) to be bridged (configured as *inclusive*). No *Special 16th character* is configured.

```
NETBIOS Name boston config>ADD INCLUSIVE ASCII
Hostname[]?westboro
Special 16th character in ASCII hex (<CR> for no special character)[]?
NETBIOS Name boston config>
```

If you do not want to be prompted, enter all parameters as one string on the command line. Use a space between each parameter.

4. Verify the filter item entry.

   Enter **LIST** to verify your entry.

```
NETBIOS Name boston config>LIST

NAME Filter List Name: boston
NAME Filter List Default: Inclusive

 Item #   Type    Inc/Ex    Hostname         Last Char

   1      ASCII    Inc      westboro

NETBIOS Name boston config>
```

5. Add additional filter items to filter list

   Repeat **step 3** to add filter items to the filter list.

   The order in which you enter filter items is important. This determines how the router applies the filter items to a packet. This first match stops the application of filter items and the router either forwards or drops the packets, depending on whether the filter item is *Inclusive* or *Exclusive*.

   Entering the most common filter items first makes the filtering process more efficient because the software is more likely to make a match at the beginning of the list.

   If the packet does not match any of the filter items, the router uses the default condition (*Inclusive* or *Exclusive*) of the filter list. You can change the default condition of the list by entering **DEFAULT INCLUSIVE** or **DEFAULT EXCLUSIVE** at the filter list configuration prompt. For example:

```
NETBIOS Name boston config>DEFAULT EXCLUSIVE
```

6. When you finish adding filter items to the filter list, enter **EXIT** to return to the `NetBIOS Filter config>` prompt.

```
NETBIOS Name boston config>EXIT
NETBIOS Filter config>
```

7. Add the filter list to your configuration.

   Use the **FILTER-ON** command. When you turn on a name filter, you must specify the following parameters in this order.

- *Input* filters incoming packets or *output* filters outgoing packets.
- *Port Number* is the desired configured bridging port number on the router.
- *Filter-list* is the name of the filter list (containing filter items) that you want to be included in this filter.
- Optionally add additionally filters list to the filter. Enter **AND** or **OR** in upper-case letters followed by a filter list name.

The following example adds a name filter comprised of the name filter list boston. The router evaluates all packets input on port 2 according to the filter items in the filter list boston. This means the router bridges all packets input on port 3 that contain the name westboro.

```
NETBIOS Filter config>FILTER-ON INPUT
Port Number[1]? 2
Filter List[]?boston
Operator (AND or OR)[]?
NETBIOS Filter config>
```

Another example:

```
NETBIOS Filter config>FILTER-ON OUTPUT
Port Number[1]?
Filter List[]?boston
Operator (AND or OR)[]?OR
Filter List[]?newyork
Operator (AND or OR)[]?
NETBIOS Filter config>
```

8. Enter **LIST** to verify the new filter.

```
NETBIOS Filter config>LIST

NETBIOS Filtering: Disabled

NETBIOS Filter Lists
--------------------

    Handle           Type

    boston           Name
    newyork          Name

NETBIOS Filters
---------------

    Port #      Direction      Filter List Handle(s)

       2          Input        boston
       1          Output       boston OR newyork

NETBIOS Filter config>
```

9. Globally enable NetBIOS name and byte filtering on the outer.
   Enter **ENABLE NETBIOS-FILTERING**.

```
NETBIOS Filter config>ENABLE NETBIOS-FILTERING
NETBIOS Filter config>
```

Example 2: Creating a Byte Filter

Use the following procedure as a guideline for creating a byte filter. Before you begin, display the `NetBIOS Filter config>` prompt.

```
Config>PROTOCOL ASRT

-- ASRT Bridge user configuration --
ASRT config>NETBIOS

NetBIOS Support User Configuration

NetBIOS config>SET FILTERS BYTE
NETBIOS Filtering configuration
NETBIOS Filter config>
```

1. Create an empty byte filter list

   Use the **CREATE BYTE-FILTER-LIST** command.

```
NETBIOS Filter config>CREATE BYTE-FILTER-LIST
Handle for Byte Filter List[]?westport
NETBIOS Filter config>
```

2. Display the configuration prompt for the filter list you just created.

   Enter **UPDATE**. The router prompts you for the name of the filter list.

```
NETBIOS Filter config>UPDATE
Handle for Filter List[]?westport
Byte Filter List Configuration
NETBIOS Byte westport config>
```

3. Add filter items to the byte filter list.

   When you add a filter item, you must specify the following parameters in this order:

   - *Inclusive* (bridged) or *exclusive* (dropped).
   - *Byte offset* is the number of bytes (in decimal) to offset into the packet the router is filtering. This starts at the NetBIOS header of the packet. Zero specifies that the router examines all bytes in the packet.
   - *Hex pattern* is a hexadecimal number the router used to compare with the bytes starting at the byte offset. See the **"NetBIOS Name and Byte Filter Configuration Commands"** and **"NetBIOS Name and Byte Filter Monitoring Commands"** sections for the syntax rules.
   - *Hex mask* if present, must be the same length as hex pattern. It is logically ANDed with the bytes in the packet, starting at byte offset, before the router compares the result with the hex pattern. If you omit the hex mask, the router considers it to be all binary 1s.

   The following example adds a filter item to the byte filter list westboro that causes the router to bridge packets with a hex pattern 0x12345678 at a byte offset of 0 (configured as *inclusive*). No hex mask is present.

```
NETBIOS Byte westport config>ADD INCLUSIVE
Byte Offset[0]?
Hex Pattern[]?12345678
Hex Mask (<CR> for no mask)[]?
NETBIOS Byte westport config>
```

4. Verify the filter item entry with the **LIST** command.

```
NETBIOS Byte westport config>LIST

BYTE Filter List Name: westport
BYTE Filter List Default: Inclusive

 Item #   Inc/Ex   Offset  Pattern               Mask

   1       Inc        0    0x12345678            0xffffffff

NETBIOS Byte westport config>
```

5. Add additional filter items to the filter list

   Repeat **step 3** to add filter items to the filter list.

   The order in which you enter filter items is important. This determines how the router applies the filter to a packet. The first match stops the application of filter items and the router either forwards or drops the packet, depending on whether the filter is *Inclusive* or *Exclusive*.

   Entering the most common filter items first makes the filtering process more efficient because the software is more likely to make a match at the beginning of the list rather than having to check the whole list before making a match.

   If the packet does not match any of the filter items, the router uses the default condition (*Inclusive* or *Exclusive*) of the filter list. You can change the default condition of the list by entering **DEFAULT INCLUSIVE** or **DEFAULT EXCLUSIVE** at the filter list configuration prompt. For example:

```
NETBIOS Byte westport config>DEFAULT EXCLUSIVE
NETBIOS Byte westport config>
```

6. When you have finished adding filter items to the list, enter **EXIT** to return to the `NetBIOS Filter config>` prompt.

```
NETBIOS Byte westport config>EXIT
NETBIOS Filter config>
```

7. Add the filter to your configuration.

   Use the **FILTER-ON** command. When you turn on a byte filter, you must specify the following parameters in this order:

   - *Input* filters incoming packets or *output* filters outgoing packets.
   - *Port Number* is the desired configured bridging port number.
   - *Filter list* is the name of the filter list (containing filter items) that you want included in this filter.
   - Optionally add additional filter lists to the filter. Enter **AND** or **OR** in upper-case letters followed by a filter list name.

The following example adds a byte filter to packets output on port 3. It is comprised of the byte filter list westboro. The router evaluates all packets output on port 3 according to filter items contained in the filter list westboro.

```
NETBIOS Filter config>FILTER-ON OUTPUT
Port Number[1]? 3
Filter List[]?westport
Operator (AND or OR)[]?
NETBIOS Filter config>
```

8. Verify the new filter.

   Enter **LIST** to verify the filter.

```
NETBIOS Filter config>LIST

NETBIOS Filtering: Enabled

NETBIOS Filter Lists
--------------------

    Handle           Type

    boston           Name
    newyork          Name
    westport         Byte

NETBIOS Filters
---------------

    Port #      Direction      Filter List Handle(s)

       2          Input        boston
       1          Output       boston OR newyork
       3          Output       westport

NETBIOS Filter config>
```

9. Globally enable NetBIOS name and byte filtering on the router.

   Enter **ENABLE NETBIOS-FILTERING**.

```
NetBIOS Filter config>ENABLE NETBIOS-FILTERING
```

3. NetBIOS Name and Byte Filter Configuration Commands

Table 11.1 lists the NetBIOS name and byte filtering configuration commands

Table 11.1. NetBIOS Name and Byte Filter configuration commands

Command	Function
? (HELP)	Lists available commands or options.
CREATE	Creates byte filter and name filter lists for NetBIOS filtering.
DELETE	Deletes byte filter and name filter lists for NetBIOS filtering.
DISABLE	Disables NetBIOS name and byte filtering on the router.
ENABLE	Enables NetBIOS name and byte filtering on the router.
FILTER-ON	Assigns a filter to a specific port. You can then apply this filter to NetBIOS packets input or output on the specified port.
LIST	Displays all information concerning created filters.
UPDATE	Adds information to or deletes information from a name or byte filter list.
EXIT	Returns you to the previous prompt.

3.1. ? (HELP)

Lists available commands or options.

Syntax:

```
NETBIOS Filter config>?
```

Example:

```
NETBIOS Filter config>?
CREATE
DELETE
DISABLE
ENABLE
FILTER-ON
LIST
UPDATE
EXIT
NETBIOS Filter config>
```

3.2. CREATE

Creates a byte filter list or a name filter list.

Syntax:

```
NETBIOS Filter config>CREATE ?
BYTE-FILTER-LIST
NAME-FILTER-LIST
```

a) CREATE BYTE-FILTER-LIST

Creates a byte filter list. Give the list a unique name of up to 16 characters. You use this name to identify the filter list.

Example:

```
NETBIOS Filter config>CREATE BYTE-FILTER-LIST
Handle for Byte Filter List[]? westport
NETBIOS Filter config>
```

b) CREATE NAME-FILTER-LIST

Creates a name filter list. Give the list a unique name of up to 16 characters. You use this name to identify the filter list.

Example:

```
NETBIOS Filter config>CREATE NAME-FILTER-LIST
Handle for Name Filter List[]? newyork
NETBIOS Filter config>
```

3.3. DELETE

Deletes byte filter lists, host name filter lists, and filters. **DELETE** removes all information associated with byte and host-name filter lists.

Syntax:

```
NETBIOS Filter config>DELETE ?
FILTER
BYTE-FILTER-LIST
NAME-FILTER-LIST
```

a) DELETE FILTER

Syntax:

```
NETBIOS Filter config>DELETE FILTER ?
INPUT
OUTPUT
```

· DELETE FILTER INPUT

Deletes a filter created with the **FILTER-ON INPUT** command.

Removes all information associated with the filter and fills any resulting gap in filter numbers.

Example:

```
NETBIOS Filter config>DELETE FILTER INPUT
Port Number[1]? 2
NETBIOS Filter config>
```

· DELETE FILTER OUTPUT

Deletes a filter created with the **FILTER-ON OUTPUT** command.

Removes all information associated with the filter and fills any resulting gap in filter numbers.

Example:

```
NETBIOS Filter config>DELETE FILTER OUTPUT
Port Number[1]? 3
NETBIOS Filter config>
```

b) DELETE BYTE-FILTER-LIST

Deletes a byte filter list

Example:

```
NETBIOS Filter config>DELETE BYTE-FILTER-LIST
Handle for Byte Filter List[]? seattle
NETBIOS Filter config>
```

c) DELETE NAME-FILTER-LIST

Deletes a host-name filter list.

Example:

```
NETBIOS Filter config>DELETE NAME-FILTER-LIST
Handle for Name Filter List[]? alaska
NETBIOS Filter config>
```

3.4. DISABLE

Globally disables NetBIOS name and byte filtering on the router.

Syntax:

```
NETBIOS Filter config>DISABLE NETBIOS-FILTERING
```

Example:

```
NETBIOS Filter config>DISABLE NETBIOS-FILTERING
NETBIOS Filter config>
```

3.5. ENABLE

Globally enables NetBIOS name and byte filtering on the router.

Syntax:

```
NETBIOS Filter config>ENABLE NETBIOS-FILTERING
```

Example:

```
NETBIOS Filter config>ENABLE NETBIOS-FILTERING
NETBIOS Filter config>
```

3.6. FILTER-ON

Assigns one or more previously configured filter lists to the input or output a specific port.

Syntax:

```
NETBIOS Filter config>FILTER-ON ?
INPUT
OUTPUT
```

a) *FILTER-ON INPUT*

Assigns one or more filter lists to incoming packets on a port. The router applies the resulting filter to all NetBIOS packets input on the specified port.

Port# is a configured bridging port number on the router. The port number identifies this filter. Enter **LIST** to see a list of port numbers. Use the **CREATE** command to make a filter list. To add additional filter lists to this port, enter **AND** or **OR** in all capital letters followed by the filter list name.

The router applies the filter you create with this command to all incoming NetBIOS packets on the specified port. The router evaluates each filter list on the command line from left to right. If a packet matches an *inclusive* filter the router bridges the packet. If a packet matches an *exclusive* filter, the router drops the packet.

If the packet is not one of the types that NetBIOS name or byte filtering supports, the router bridges the packet.

Example:

```
NETBIOS Filter config>FILTER-ON INPUT
Port Number[1]? 2
Filter List[]? boston
Operator (AND or OR)[]?
NETBIOS Filter config>
```

b) *FILTER-ON OUTPUT*

Assigns one or more filter lists to outgoing packets on a port. The router applies the resulting filter to all NetBIOS packets output on the specified port.

Port# is a configured bridging port number on the router. The port number identifies this filter. Enter **LIST** to see a list of port numbers. Use the **CREATE** command to make a filter list. To add additional filter lists to this port, enter **AND** or **OR** in all capital letters followed by the filter list name.

The router applies the filter you create with this command to all outgoing NetBIOS packets on the specified port. The router evaluates each filter list on the command line from left to right. If a packet matches an *inclusive* filter the router bridges the packet. If a packet matches an *exclusive* filter, the router drops the packet.

If the packet is not one of the types that NetBIOS name or byte filtering supports, the router bridges the packet.

Example:

```
NETBIOS Filter config>FILTER-ON OUTPUT
Port Number[1]?
Filter List[]? boston
Operator (AND or OR)[]? OR
Filter List[]? newyork
Operator (AND or OR)[]?
NETBIOS Filter config>
```

3.7. LIST

Displays information on all name and byte filters.

Syntax:

```
NETBIOS Filter config>LIST
```

Example:

```
NETBIOS Filter config>LIST

NETBIOS Filtering: Enabled

NETBIOS Filter Lists
--------------------

    Handle           Type

    boston           Name
    newyork          Name
    westport         Byte

NETBIOS Filters
---------------

    Port #      Direction      Filter List Handle(s)

       2          Input        boston
       1          Output       boston OR newyork
       3          Output       westport

NETBIOS Filter config>
```

NetBIOS Filtering Displays whether NetBIOS filtering is enabled or disabled.

NetBIOS Filter Lists Shows the name (handle) of the filter lists, as well as the type, either Name or Byte.

NetBIOS Filters Assigned port number and direction (input or output) of each filter. Filter List Handle(s) displays the name(s) of the filter list(s) making up the filter.

3.8. UPDATE

Displays the `NETBIOS Byte (or Name) filter-list config>` prompt, which lets you update the specified filter list. At this prompt you can add, delete, list, or move items in byte and name filter lists. You can also set the default of each filter list to *inclusive* or *exclusive*.

Syntax:

```
NETBIOS Filter config>UPDATE <filter-list>
```

Example:

```
NETBIOS Filter config>UPDATE
Handle for Filter List[]? newyork
Name Filter List Configuration
NETBIOS Name newyork config>
```

At this new prompt, you can enter several commands.

3.9. EXIT

Returns to the previous prompt.

Syntax:

```
NETBIOS Filter config>EXIT
```

Example:

```
NETBIOS Filter config>EXIT
NETBIOS config>
```

4. NetBIOS Name and Byte Filter Monitoring Commands

Table 11.2 lists the NetBIOS name and byte filtering monitoring commands

Table 11.2. NetBIOS Name and Byte Filter Monitoring commands

Command	**Function**
? (HELP)	Lists available commands or options.
LIST	Displays all information concerning created filters.
EXIT	Returns you to the previous prompt.

4.1. ? (HELP)

Lists available commands or options.

Syntax:

```
NETBIOS Filter>?
```

Example:

```
NETBIOS Filter>?
LIST
EXIT
NETBIOS Filter>
```

4.2. LIST

Displays information on all filters, on bytes, or on name filters.

Syntax:

```
NETBIOS Filter>LIST ?
BYTE-FILTER-LISTS
NAME-FILTER-LISTS
FILTERS
```

a) LIST BYTE-FILTER-LISTS

Displays all of the byte filter list that you created.

Example:

```
NETBIOS Filter>LIST BYTE-FILTER-LIST

BYTE Filter List Name: westport
BYTE Filter List Default: Exclusive

 Filter Item #   Inc/Ex    Byte Offset  Pattern       Mask

      1        Inclusive       0        0x12345678    0xffffffff

NETBIOS Filter>
```

b) LIST NAME-FILTER-LISTS

Displays all of the name filter lists that you created.

Example:

```
NETBIOS Filter>LIST NAME-FILTER-LISTS

NAME Filter List Name: boston
NAME Filter List Default: Inclusive

 Filter Item #   Type    Inc/Ex        Hostname       Last Char

       1         ASCII   Inclusive     westboro
       2         ASCII   Inclusive     seattle


NAME Filter List Name: newyork
NAME Filter List Default: Inclusive

 Filter Item #   Type    Inc/Ex        Hostname       Last Char

       1         ASCII   Inclusive     jersey

NETBIOS Filter>
```

c) LIST FILTERS

Lists all of the filters that you created and the number of packets the router filtered as a result of those filters.

Example:

```
NETBIOS Filter>LIST FILTERS

NETBIOS Filtering: Enabled

    Port #      Direction      Filter List Handle(s)   Pkts Filtered

       2          Input        boston                         0
       1          Output       boston OR newyork              0
       3          Output       westport                       0

NETBIOS Filter>
```

4.3. EXIT

Returns to the previous prompt.

Syntax:

```
NETBIOS Filter>EXIT
```

Example:

```
NETBIOS Filter>EXIT
NETBIOS>
```

5. Update Byte-Filter-List Commands

This section describes the commands available at the `NETBIOS Byte filter-list config>` prompt.

add *inclusive* or *exclusive byte-offset hex-pattern hex-mask*

Adds a filter item to the filter list. When you add a filter item, the router numbers the item and displays the number of the filter item you just added.

> ***Note: Adding filter items to filter lists adds to processing time due to the time it takes to evaluate each item in the list. It can affect performance in heavy NetBIOS traffic.***

The order in which you enter filter items is important as this determines how the router applies filter items to a packet. The router stops comparing the packet to a filter when it finds the first match.

- *Inclusive* (bridged) or *exclusive* (dropped).
- *Byte offset* is the number of bytes (in decimal) to offset into the packet the router is filtering. This starts at the NetBIOS header of the packet. Zero specifies that the router examines all bytes in the packet.
- *Hex pattern* is a hexadecimal number the router used to compare with the bytes starting at the byte offset. Syntax rules for *hex-pattern* include no 0x in front, a maximum of 32 numbers, and an even number of hex numbers.
- *Hex mask* if present, must be the same length as hex pattern. It is logically **AND**ed with the bytes in the packet, starting at byte offset, before the router compares the result with the hex pattern. If you omit the hex mask, the router considers it to be all binary 1s.

If the offset and pattern of a byte filter item represent bytes that do not do not exist in a NetBIOS packet (for example, if the packet is shorter than was intended when setting up a byte-filter list), the router does not apply the filter to the packet. If you use a series of byte filter items to set up a single NetBIOS filter list, then a packet is not tested for filtering if any of the byte filter items within the NetBIOS filter list represent bytes that do not exist in the NetBIOS packet.

Example:

```
NETBIOS Byte westport config>ADD INCLUSIVE
Byte Offset[0]?
Hex Pattern[]? 12345678
Hex Mask (<CR> for no mask)[]?
NETBIOS Byte westport config>
```

The following example shows how to filter Datagram Broadcast Packets.

Example:

```
NETBIOS Byte westport config>ADD INCLUSIVE
Byte Offset[0]? 4
Hex Pattern[]? 09
Hex Mask (<CR> for no mask)[]?
NETBIOS Byte westport config>
```

default inclusive or exclusive

Changes the default setting of the filter list to *inclusive* or *exclusive*. If no filter items match the contents of the packet the router considers for filtering, the router forwards or drops the packet, depending on this setting.

Syntax:

```
NETBIOS Byte filter-list config>DEFAULT
INCLUSIVE
EXCLUSIVE
```

Example 1:

```
NETBIOS Byte westport config>DEFAULT INCLUSIVE
NETBIOS Byte westport config>
```

Example 2:

```
NETBIOS Byte westport config>DEFAULT INCLUSIVE
NETBIOS Byte westport config>
```

delete filter-number

Deletes a filter item from the filter list. The software immediately renumbers the list. To see a list of item numbers, enter **LIST**.

Syntax:

```
NETBIOS Byte filter-list config>DELETE <filter #>
```

Example:

```
NETBIOS Byte westport config>DELETE
Filter Item Number[1]? 2
NETBIOS Byte westport config>
```

exit

Returns to the previous command prompt level.

Syntax:

```
NETBIOS Byte filter-list config>EXIT
```

Example:

```
NETBIOS Byte westport config>EXIT
NETBIOS Filter config>
```

list

Displays information related to filter items in the filter list.

Syntax:

```
NETBIOS Byte filter-list config>LIST
```

Example:

```
NETBIOS Byte westport config>LIST

BYTE Filter List Name: westport
BYTE Filter List Default: Inclusive

 Item #   Inc/Ex   Offset  Pattern               Mask

   1       Inc        4    0x09                  0xff
   2       Ex         2    0x3344                0xffff

NETBIOS Byte westport config>
```

move *filter-item-1 filter-item-2*

Re-orders filter items within the filter list. To see a list of item numbers, enter **LIST**.

Syntax:

```
NETBIOS Byte filter-list config>MOVE <filter-item-1, filter-item-2>
```

Example:

```
NETBIOS Byte filter-list config>MOVE
Source Filter Item Number [1] ?  3
After Destination Filter Item Number [0] ?  1
NETBIOS Byte filter-list config>
```

6. Update Name-Filter-List Commands

This section lists the commands available at the `NETBIOS Name filter-list config>` prompt.

Available commands are:

- ADD
- DEFAULT
- DELETE
- EXIT
- LIST
- MOVE

add *inclusive* or *exclusive* ASCII *host-name special-16th-char*

Adds a filter item to the name filter list. The router compares the following frames and fields with the information you enter with this command:

- ADD_GROUP_NAME_QUERY: Source NetBIOS name field
- ADD_NAME_QUERY: Source NetBIOS name field
- DATAGRAM: Destination NetBIOS name field
- NAME_QUERY: Destination NetBIOS name field

Enter the following information with this command:

- *Inclusive* (bridged) or *exclusive* (dropped).
- *Hostname* is an ASCII string up to 16 characters. It can contain any character but the following: **. / \ [ ] : ‖ < > + = ; ,** space. Use ? to indicate a single character wildcard. Use * as the final character of the name to indicate a wildcard for the remainder of the name. If the name contains fewer than 15 characters, it is padded to the 15th character with ASCII spaces.
- *Special 16th character* can be used if *host-name* contains fewer than 16 characters. It is a hexadecimal number (with no 0x in front of it) that indicates the value for the last character. If you do not specify a 16th character on a name less than 16 characters, the router uses a ? wildcard for the 16th character.

Example:

```
NETBIOS Name boston config>ADD INCLUSIVE ASCII
Hostname[]? qwerty
Special 16th character in ASCII hex (<CR> for no special character)[]?
NETBIOS Name boston config>
```

add *inclusive* or *exclusive* HEX hexstring

Adds a filter item to the name filter. This command is functionally the same as **ADD INCLUSIVE ASCII**. However, you enter the name as a series of hexadecimal numbers (with no 0x in front).

Hex string must consist of an even number of hexadecimal numbers. Specify a wildcard for a single byte by ??. If you do not supply a full 32 hexadecimal numbers, the router pads ASCII blanks to the 29th and 30th numbers and supplies a wildcard as the 31st and 32nd (16th byte) numbers.

Example:

```
NETBIOS Name boston config>ADD EXCLUSIVE HEX
Hex String[]? abc123987fed
NETBIOS Name boston config>
```

default *inclusive* or *exclusive*

Changes the default setting of the filter list to *inclusive* or *exclusive*. If no filter items match the packet the router considers for filtering, the router forwards or drops the packet, depending on this setting.

Syntax:

```
NETBIOS Name filter-list config>DEFAULT ?
INCLUSIVE
EXCLUSIVE
```

Example:

```
NETBIOS Name filter-list config>DEFAULT INCLUSIVE
NETBIOS Name filter-list config>
```

delete *filter-item*

Deletes a filter item from the list. To see a list of item numbers, enter **LIST**.

Syntax:

```
NETBIOS Name filter-list config>DELETE <filter #>
```

Example:

```
NETBIOS Name filter-list config>DELETE
Filter Item Number [1] ? 4
NETBIOS Name filter-list config>
```

exit

Exits to the previous prompt level.

Syntax:

```
NETBIOS Name filter-list config>EXIT
```

Example:

```
NETBIOS Name filter-list config>EXIT
NETBIOS Filter config>
```

list

Displays information related to items in the specified filter list.

Syntax:

```
NETBIOS Name filter-list config>LIST
```

Example:

```
NETBIOS Name boston config>LIST

NAME Filter List Name: boston
NAME Filter List Default: Inclusive

 Item #   Type    Inc/Ex    Hostname         Last Char

   1      ASCII    Inc      westboro
   2      ASCII    Inc      seattle
   3      HEX      Ex       abc123987fed

NETBIOS Name boston config>
```

move *filter-item 1 filter-item 2*

Re-orders filter items within the filter list. To see a list of item numbers enter **LIST**.

Syntax:

```
NETBIOS Name filter-list config>MOVE <nº origen, nº final>
```

Example:

```
NETBIOS Name boston config>MOVE
Source Filter Item Number[1]? 1
After Destination Filter Item Number[0]? 3
NETBIOS Name boston config>
```

Chapter 12
Using MAC Filtering

1. About MAC Filtering

MAC filtering lets you set up packet filters. Filters are a set of rules applied to a packet to determine how it is handled.

> ***Note: MAC filtering is allowed on tunnel traffic.***

During the filtering process, packets are either processed, filtered, or tagged.

- *Processed* - Packets are permitted to pass through the bridge unaffected.
- *Filtered* - Packets are not permitted to pass through the bridge.
- *Tagged* - Packets are allowed to pass through the bridge but are marked with a number in the range of 1 to 64 based on a configurable parameter.

A MAC filter is made up of three objectives:

- *Filter-item* - A single rule for the address field of a packet. The result is either TRUE (the match was successful or FALSE (the match was not successful).
- *Filter-list* - Contains a list of one or more filter-items.
- *Filter* - Contains a set of filter-lists.

1.1. MAC Filtering and DLSw Traffic

You can set up MAC filtering to channel eligible DLSw traffic to alternate bridge paths on a MAC station basis.

To set up a filter for LLC, use the *Bridge Net* as the interface number for the filter. Calculate the Bridge Net number by adding two to the number of interfaces configured for your router. Enter **LIST DEVICES** at the `Config>` prompt or enter **CONFIGURATION** at the **+** prompt to see a list of interfaces.

When you set up a filter for the Bridge Net, for example, the router does not drop frames that match exclusive filters. Instead, it forwards those frames to the bridge.

2. Using MAC Filtering Parameters

You can specify some or all of the following parameters when you create a filter.

- Source MAC address or destination MAC address
- Mask to be applied to the packet's fields to be filtered
- Interface number
- Input/output designation
- Include/exclude/tag designation
- Tag value (if you designate a tag)

2.1. Filter-Item Parameters

You specify the following parameters to construct a filter-item.

- Address Type: *source or destination*
- Tag: *Tag-value*
- Address Mask: *Hex-Mask*

Each filter-item specifies an address type (source or destination to match against the type in the packet with the tokens.

The *address mask* is a MAC address in hex comparing the packet's addresses. The mask is applied to the source destination MAC address of the packet before comparing it against the specified MAC address.

The mask specifies the bytes that are to be logically ANDed with the bytes in the MAC address. It must be of equal length to the specified MAC address. If no mask is specified, it is assumed to be all 1's.

2.2. Filter List Parameters

The following parameters are used to construct a filter list:

- Name: *ASCII-string*
- Filter-item List: *filter-item 1, …, filter-item n*
- Action: INCLUDE, EXCLUDE, TAG (n)

A filter list is built from one or more filter items. Each filter list is given a unique name.

Applying a filter list to a packet consists of comparing each filter item in the order by which the filter item were added to the list. If any of the filter items in the list return TRUE then the filter list returns its designated action.

2.3. Filter Parameters

The following parameters are used to construct a filter:

- Filter list Names: *ASCII-string, …, ASCII string*
- Interface Number: *IFC-number*
- Port Direction: *input* or *output*
- Default Action: *include, exclude,* or *tag*
- Default Tag: *tag value*

A *filter* is constructed by associating a group of filter names with an interface number and assigning an input or output designation. The application of a filter to a packet means that each of the associated filter lists should be applied to packets being received (input) or sent (output) on the specified interface.

When a filter evaluates a packet to an *include* condition, the packet is forwarded. When a filter evaluates a packet to an *exclude* condition, the packet is dropped. When a filter evaluates to a *tag* condition, the packet being considered is forwarded with a tag.

An additional parameter of each filter is the default action which is the result of non-match for all of its filter lists. This default action is include. It can be set to either include, exclude, or tag. In addition, if the default action is tag, a tag value is also given.

3. Using MAC Filtering Tags

- MAC Address filtering is handled by a joint effort between bandwidth reservation and the MAC filtering feature (MCF) using *tags*. A user with bandwidth reservation is able to categorize bridge traffic, for example, by assigning a tag to it.
- Tagging is done by creating a filter item at the MAC filtering configuration prompt and assigning a tag to it. This tag is used to set up a bandwidth class for all packets associated with this tag. Tag values must be in the range of 1 to 64.

  Supports applying tags only to bridged packets and allows only the MAC address fields of the packet to be used in applying the tag.
- Up to five tagged MAC addresses can be set from 1 to 5. TAGI will be searched for first, then TAG2, and so on.
- Once a tagged filter is created, it is assigned a class and priority in the Bandwidth Reservation configuration process. Use **TAG** at the Bandwidth Reservation to reference the tag.

Tags can also refer to groups as in IP Tunnel. Tunnel end points can belong to any number of groups, and then packets are assigned to a particular group through the tagging feature of MAC address filtering.

Chapter 13
Configuration and Monitoring MAC Filtering

1. Accessing the MAC Filtering Prompts

To display the MAC filtering configuration prompt, at the `Config>` prompt enter **FEATURE MAC-FILTERING**. For example:

```
Config>FEATURE MAC-FILTERING

-- MAC Filtering user configuration --
Filter Config>
```

To display the MAC filtering configuration prompt, at the + prompt enter **FEATURE MAC-FILTERING**. For example:

```
+FEATURE MAC-FILTERING

-- MAC Filtering user console --
Filter>
```

2. MAC Filtering Configuration Commands

This section describes the MAC filtering configuration commands. Enter configuration commands at the `Filter config>` prompt. Table 13.1 lists the MAC filtering commands.

Table 13.1 MAC Filtering Commands.

Command	**Function**
? (HELP)	Displays available commands or options.
ATTACH	Adds a filter list to a filter.
CREATE	Creates a filter list or an *input* or *output* filter.
DEFAULT	Sets the default action for the filter with a specified *filter-number* to *exclude, include* or *tag*.
DELETE	Removes all information associated with a filter list and frees an assigned string as a name for a new filter list. Also deletes a filter created.
DETACH	Deletes a filter-list name from a filter.
DISABLE	Disables MAC filtering globally or on a per filter basis.
ENABLE	Enables MAC filtering globally or on a per filter basis.
LIST	Lists a summary of statistics and settings for each filter currently running in the router.
MOVE	Reorders the filter-lists attached to a specific filter.
REINIT	Re-initializes the entire MAC filtering system without affecting the rest of the router.
SET-CACHE	Changes the cache size for a filter.
UPDATE	Adds or deletes information from a filter-list. Brings you to a menu of appropriate subcommands.
EXIT	Exits the MAC filtering configuration process.

2.1. ? (HELP)

Lists available commands or options.

Syntax:

```
Filter Config>?
```

Example:

```
Filter Config> ?
ATTACH
CREATE
DEFAULT
DELETE
DETACH
DISABLE
ENABLE
LIST
MOVE
```

```
REINIT
SET-CACHE
UPDATE
EXIT
```

2.2. ATTACH

Adds a filter list to a filter. A filter is constructed by associating a group of filter lists with an interface number. A filter list is built from one or more filter items.

Syntax:

```
Filter Config>ATTACH <filter-list-name, filter-number>
```

Example:

```
Filter Config>ATTACH
Enter a filter-list name[]? paris
Enter a filter number[1]?
Filter Config>
```

2.3. CREATE

Creates a filter list or an input or output filter.

Syntax:

```
Filter Config>CREATE ?
LIST
FILTER
```

a) CREATE LIST

Creates a filter list. Name a list by a unique string (*Filter-list-name*) of up to 16 characters. This name is used to identify a filter-list that is being built. This name is also used with other commands associated with the filter-list.

Example:

```
Filter Config>CREATE LIST
Enter a filter-list name[]? probe-list
Filter Config>
```

b) CREATE FILTER

Creates a filter and places it on the network associated with the *input* or *output* direction on the interface given by an interface number. By default this filter is created with no attached filter-lists and has a default action of *include* and *enabled*.

Example:

```
Filter Config>CREATE FILTER
Enter a direction to filter (INPUT or OUTPUT)[INPUT]?
Enter an interface to filter[0]? 2
Filter Config>
```

2.4. DEFAULT

Sets the default action for the filter with a specified *filter-number* to *exclude, include*, or *tag*.

Syntax:

```
Filter Config>DEFAULT ?
EXCLUDE
INCLUDE
TAG
```

a) DEFAULT EXCLUDE

Sets the default action for the filter with a specified *filter-number* to *exclude*.

Example:

```
Filter Config>DEFAULT EXCLUDE
Enter a filter number[1]? 2
Filter Config>
```

b) DEFAULT INCLUDE

Sets the default action for the filter with a specified *filter-number* to *include*.

Example:

```
Filter Config>DEFAULT INCLUDE
Enter a filter number[1]? 3
Filter Config>
```

c) DEFAULT TAG

Sets the default action for the filter with a specified *filter-number* to *tag* and sets the associated tag value to *tag-number*.

Example:

```
Filter Config>DEFAULT TAG
Enter a tag value[1]? 2
Enter a filter number[1]? 1
Filter Config> CONFIG>
```

2.5. DELETE

Removes all information associated with a filter-list and frees an assigned string as a name for a new filter-list. If filter-list is attached to a filter that has already been created, then this command displays an error message without deleting anything. In addition all filter-items belonging to this list are also deleted.

This command also deletes a filter created using the **CREATE FILTER** command.

Syntax:

```
Filter Config>DELETE ?
LIST
FILTER
```

a) DELETE LIST

Removes all information associated with a filter-list and frees an assigned string as a name for a new filter-list. The filter-list must be a string entered by a previous **CREATE LIST** command.

If the filter-list is attached to a filter that has already been created, then this command displays an error message without deleting anything. All filter-items belonging to this list are also deleted when this command is used.

Example:

```
Filter Config>DELETE LIST
Enter a filter-list name[]? probe-list
Filter Config>
```

b) DELETE FILTER

Deletes a filter created using the **CREATE FILTER** command.

Example:

```
Filter Config>DELETE FILTER
Enter a filter number[1]? 1
Filter Config>
```

2.6. DETACH

Deletes a filter-list name (*filter-list* parameter) from a filter (*filter-number* parameter).

Syntax:

```
Filter Config>DETACH <filter-list-name, filter-number>
```

Example:

```
Filter Config>DETACH
Enter a filter-list name[]? paris
Enter a filter number[1]? 2
Filter Config>
```

2.7. DISABLE

Disables MAC filtering entirely or disables a particular filter.

Syntax:

```
Filter Config>DISABLE ?
ALL
FILTER
```

a) DISABLE ALL

Disables MAC filtering entirely. Filters are still set as *enabled*, however, if they were enabled previously.

Example:

```
Filter Config>DISABLE ALL
Filter Config>
```

b) DISABLE FILTER

Disables a particular filter. The *filter number* parameter corresponds to the numbers displayed with the **LIST FILTERS** command.

Example:

```
Filter Config>DISABLE FILTER
Enter a filter number[1]? 2
Filter Config>
```

2.8. ENABLE

Enables MAC filtering entirely or enables a particular filter.

Syntax:

```
Filter Config>ENABLE ?
ALL
FILTER
```

a) *ENABLE ALL*

Enables MAC filtering entirely although filters themselves may still set to disabled.

Example:

```
Filter Config>ENABLE ALL
Filter Config>
```

b) *ENABLE FILTER*

Enables a particular filter. The *filter number* parameter corresponds to the numbers displayed with the **LIST FILTERS** command.

Example:

```
Filter>ENABLE FILTER
Enter a filter number[1]? 1
Filter>
```

2.9. LIST

Syntax:

```
Filter Config>LIST ?
ALL
FILTER
```

a) *LIST ALL*

Lists all the filter lists and filters that you have configured. A list of all the filter lists attached to a filter is not given. Other information displayed includes:

- Whether or not filtering is enabled or disabled
- A list containing the state of the filtering system (enable, disable)
- The set of configured filter-list records
- Each of the configured filter records

In addition, the following information is displayed for each filter:

- Filter number
- Interface number
- Filter direction (input, output)
- Filter state (enable, disable)
- Filter default action (tag, include, exclude)

Example:

```
Filter Config>LIST ALL
Filtering: enabled
Filter List                   Action
-----------                   ------
paris                         INCLUDE

Filters
-------
Id   Default  State         Ifc  Dir       Cache
--   -------  -----         ---  ---       ------
1    INCLUDE  enabled       2    INPUT     16
2    INCLUDE  disabled      1    OUTPUT    16
3    INCLUDE  enabled       0    INPUT     16
Filter Config>
```

b) LIST FILTER

Generates a list of attached filter-lists for the specified filter and all subsequent information for the filter.

Example:

```
Filter Config>LIST FILTER
Enter a filter number[1]?
Id   Default  State         Ifc  Dir       Cache
--   -------  -----         ---  ---       ------
1    INCLUDE  enabled       2    INPUT     16

Filter List                   Action
-----------                   ------
paris                         INCLUDE
Filter Config>
```

2.10. MOVE

Use the **MOVE** command to re-order the filter-lists attached to a specified filter (given by the *filter-number* parameter). The list given by *Filter-list-name1* is moved immediately before the list given by *Filter-list-name2*.

Syntax:

```
Filter Config>MOVE <filter-list-name1, filter-list-name2, filter-number>
```

Example:

```
Filter Config>MOVE
Enter a filter-list name from[]? paris
Enter a filter-list name to[]? rome
Enter a filter number[1]? 1
Filter Config>
```

2.11. REINIT

Reinitializes the entire MAC filtering system from an existing configuration without affecting the rest of the router.

Syntax:

```
Filter Config>REINIT
```

Example:

```
Filter Config>REINIT
Reinitialize MAC Filtering? (Yes/No)? y
Filter Config>
```

2.12. SET-CACHE

Changes the cache size to a number between 4 and 32768. The default is 16.

Syntax:

```
Filter Config>SET-CACHE <filter-number, cache-size>
```

Example:

```
Filter Config>SET-CACHE
Enter a filter number[1]?
Enter the new cache size[16]? 32
Filter Config>
```

2.13. UPDATE

Use the **UPDATE** command to add information to or delete information from a specific filter-list.

Using this command with the desired *filter-list-name* brings you to the `Filter filter-list-name Config>` prompt for that filter list. From this new prompt you can change information in the list.

The order in which the filter-items are specified for a filter-list is important as it determines the order in which the filter-items are applied to a packet.

Syntax:

```
Filter Config>UPDATE <filter-list-name>
```

Example:

```
Filter Config>UPDATE PROBE
Filter 'probe' Config>
```

2.14. EXIT

Use the **EXIT** command to return to the `Config>` prompt.

Syntax:

```
Filter Config>EXIT
```

Example:

```
Filter Config>EXIT
Config>
```

3. MAC Filtering Monitoring Commands

This section describes the MAC filtering monitoring commands. Enter monitoring commands at the `Filter>` prompt. Table 13.2 lists the MAC filtering monitoring commands.

Table 13.2 MAC Filtering Commands.

Command	**Function**
? (HELP)	Displays available commands or options.
CLEAR	Clears the per filter statistics listed in the **LIST FILTER** command.
DISABLE	Disables MAC filtering globally or on a per filter basis.
ENABLE	Enables MAC filtering globally or on a per filter basis.
LIST	Lists a summary of statistics and settings for each filter currently running in the router.
REINIT	Re-initializes the entire MAC filtering system without affecting the rest of the router.
EXIT	Exits the MAC filtering configuration or monitoring process

3.1. ? (HELP)

Lists available commands or options.

Syntax:

```
Filter>?
```

Example:

```
Filter>?
CLEAR
DISABLE
ENABLE
LIST
REINIT
EXIT
```

3.2. CLEAR

Clears all the per filter statistics listed in the **LIST FILTER** command for all the filter objects and all the statistics listed for each filter list.

The command also clears the per filter statistics listed in the **LIST FILTER** command for the filter associated with the *filter-number* plus all the statistics listed for each filter list in this filter.

Syntax:

```
Filter>CLEAR ?
ALL
FILTER
```

a) CLEAR ALL

Clears all statistics listed in the **LIST FILTER** command for each filter object and each filter-list.

Example:

```
Filter>CLEAR ALL
Filter>
```

b) CLEAR FILTER

Clears the per filter statistics listed in the **LIST FILTER** command for the filter associated with the *filter-number* plus all the statistics listed for each filter-list in this filter.

Example:

```
Filter>CLEAR FILTER
Enter a filter number[1]?
Filter>
```

3.3. DISABLE

Disables MAC filtering entirely or disables a particular filter.

Syntax:

```
Filter>DISABLE ?
ALL
FILTER
```

a) DISABLE ALL

Disables MAC filtering entirely. Filters are still set as *enabled*, however, if they were enabled previously.

Example:

```
Filter Config>DISABLE ALL
Filter Config>
```

b) DISABLE FILTER

Disables a particular filter. The *filter number* parameter corresponds to the numbers displayed with the **LIST FILTERS** command.

Example:

```
Filter Config>DISABLE FILTER
Enter a filter number[1]? 2
Filter Config>
```

3.4. ENABLE

Enables MAC filtering entirely or enables a particular filter.

Syntax:

```
Filter>ENABLE ?
ALL
FILTER
```

a) ENABLE ALL

Enables MAC filtering entirely although filters themselves may still set to disabled.

Example:

```
Filter Config>ENABLE ALL
Filter Config>
```

b) ENABLE FILTER

Enables a particular filter. The *filter number* parameter corresponds to the numbers displayed with the **LIST FILTERS** command.

Example:

```
Filter>ENABLE FILTER
Enter a filter number[1]? 1
Filter>
```

3.5. LIST

Syntax:

```
Filter>LIST ?
ALL
FILTER
```

a) LIST ALL

Lists all the filter lists and filters that you have configured. A list of all the filter lists attached to a filter is not given. Other information displayed includes:

- Whether or not filtering is enabled or disabled
- A list containing the state of the filtering system (enable, disable)
- The set of configured filter-list records
- Each of the configured filter records

In addition, the following information is displayed for each filter:

- Filter number
- Interface number
- Filter direction (input, output)
- Filter state (enable, disable)
- Filter default action (tag, include, exclude)

Example:

```
Filter>LIST ALL
Filtering: enabled
Filter List                   Action
-----------                   ------
paris                         INCLUDE

Filters
-------
Id   Default  State         Ifc  Dir       Cache
--   -------  -----         ---  ---       ------
1    INCLUDE  enabled       2    INPUT     16
2    INCLUDE  disabled      1    OUTPUT    16
3    INCLUDE  enabled       0    INPUT     16
Filter>
```

b) LIST FILTER

Generates a list of attached filter-lists for the specified filter and all subsequent information for the filter.

Example:

```
Filter>LIST FILTER
Enter a filter number[1]?
Id   Default  State         Ifc  Dir       Cache
--   -------  -----         ---  ---       ------
1    INCLUDE  enabled       2    INPUT     16

Filter List                   Action
-----------                   ------
paris                         INCLUDE
Filter>
```

3.6. REINIT

Reinitializes the entire MAC filtering system from an existing configuration without affecting the rest of the router.

Syntax:

```
Filter>REINIT
```

Example:

```
Filter>REINIT
Reinitialize MAC Filtering? (Yes/No)? y
Filter Config>
```

3.7. EXIT

Use the **EXIT** command to return to the + prompt.

Syntax:

```
Filter>EXIT
```

Example:

```
Filter>EXIT
+
```

4. MAC Filtering Update Commands

Table 13.3 lists the MAC filtering update commands. Enter these commands at the `Filter 'filter-list-name' Config>` prompt.

Table 13.3 MAC Filtering Update Commands

Command	Function
? (HELP)	Displays available commands or options.
ADD	Adds a hexadecimal number to compare against the source or destination MAC address. Adds filter items to a filter list. Adds a filter list to a filter.
DELETE	Removes filter-items from a filter-list.
LIST	Lists a summary of all the filter lists and filters configured by the user. Also generates a list of attached filter lists for this filter and all subsequent information for the filter.
MOVE	Reorders the filter lists attached to a specified filter.
SET-ACTION	Sets a filter item to evaluate either *include*, *exclude* or *tag* (with a *tag-number* option).
EXIT	Exits the update subcommand configuration process.

4.1. ? (HELP)

Lists available commands or options.

Syntax:

```
Filter 'filter-list-name' Config>?
```

Example:

```
Filter 'probe' Config>?
ADD
DELETE
LIST
MOVE
SET-ACTION
EXIT
Filter 'probe' Config>
```

4.2. ADD

Adds filter-items to a filter-list. This command specifically lets you add a hexadecimal number to compare against the source or destination MAC address.

The order in which you add filter-items to a filter-list is important as it determines the order in which the filter-items are applied to a packet.

Each use of the **ADD** subcommand creates a filter-item within the filter-list. The first filter-item is assigned *filter-item-number* 1, the next one is assigned *number 2*, and so forth. After an **ADD**, the router displays the number of the filter-item just added.

The first match that occurs stops the application of filter-items, and the filter-list evaluates to either *include, exclude* or *tag*, depending on the designated action of the filter-list. If none of the filter-items of a filter-list produce a match, then the default action (*include, exclude* or *tag*) or the filter is returned.

Syntax:

```
Filter 'filter-list-name' Config>ADD ?
SOURCE
DESTINATION
```

a) ADD SOURCE

Adds a hexadecimal number (with no 0x in front, a maximum of 16 number, and an even number of hex numbers) to compare against the source MAC address.

The *hex-mask* parameter must be the same length as *hex-MAC-address* and is logically ANDed with the designed MAC address in the packet. The default *hex-mask* argument is all binary 1's.

You can enter the *hex-MAC-addr* in canonical or non-canonical bit order. Canonical bit order is just a hex number (for example, 000003001234) or a series of hex digits with a dash between every two digits (for example, 00-00-03-00-12-34).

Non-canonical bit order is a series of hex digits with a colon between every two digits (for example, 00:00:C9:09:66:49). MAC addresses of filter-items are always displayed using either dash or colon to distinguish canonical from non-canonical representations.

Example:

```
Filter 'paris' Config>ADD SOURCE
Enter MAC Address[]? 00-11-22-33-44-55
Enter MAC Mask[ffffffffffff]?
Filter 'paris' Config>
```

b) ADD DESTINATION

Acts exactly like **ADD SOURCE**, except that the match is made against the destination rather than the source MAC address of the packet.

Example:

```
Filter 'sample' Config>ADD DESTINATION
Enter MAC Address[]? 00-00-a0-bb-0f-13
Enter MAC Mask[ffffffffffff]?
Filter 'sample' Config>
```

4.3. DELETE

Removes filter-items from a filter-list. You delete filter items by specifying the filter-item-number assigned to the item when it was added.

When you delete a filter item, any gap created in the number sequence is filled in. For example, if filter-items 1.2.3 and 4 exist and you delete filter-item 3, then filter-item 4 is renumbered to 3.

Syntax:

```
Filter 'filter-list-name' Config>DELETE <filter-item-number>
```

Example:

```
Filter 'sample' Config>DELETE 2
Filter 'sample' Config>
```

4.4. LIST

Lists all the filter-item records represented in canonical and non-canonical form. It displays the following information about each filter item.

- MAC address and address mask in canonical or non-canonical form
- filter-item numbers
- address type (*source* or *destination*)
- filter-list action

Syntax:

```
Filter 'filter-list-name' Config>LIST ?
CANONICAL
NONCANONICAL
```

a) LIST CANONICAL

Lists all the filter-item records in a filter-list, giving the item numbers, the address type (SRC, DST), the MAC address in canonical form, and the address mask in canonical form. In addition gives the filter-list action.

Example:

```
Filter 'sample' Config>LIST CANONICAL
Action: INCLUDE
Id  Type  MAC Address                 Mask
--  ----  ------------                ----
1   SRC   01-02-03-04-05-06           ff-ff-ff-ff-ff-ff
2   DST   00-00-11-11-22-22           ff-ff-ff-ff-ff-ff
Filter 'sample' Config>
```

b) LIST NONCANONICAL

Lists all the filter-item records in a filter-list, giving the item numbers, the address type (SRC, DST), the MAC address in non-canonical form, and the address mask in non-canonical form. In addition gives the filter-list action.

Example:

```
Filter 'sample' Config>LIST NONCANONICAL
Action: INCLUDE
Id  Type  MAC Address                 Mask
--  ----  ------------                ----
1   SRC   80:40:c0:20:a0:60           ff:ff:ff:ff:ff:ff
2   DST   00:00:88:88:44:44           ff:ff:ff:ff:ff:ff
Filter 'sample' Config>
```

4.5. MOVE

Re-orders filter-items within the filter-list. The filter-item whose number is specified by *filter-item-name 1* is moved and renumbered to be just before *filter-item-name 2*.

Syntax:

```
Filter 'filter-list-name' Config>MOVE <filter-item-name1, filter-item-name2>
```

Example:

```
Filter 'sample' Config>MOVE
Item number to move[1]? 2
Item number before which to insert[1]?
Filter 'sample' Config>
```

4.6. SET-ACTION

Lets you set a filter-list to either *include, exclude* or *tag* (with a *tag-number* option). If one of the filter-items of the filter-list matches the contents of the packet being considered for filtering, the filter-list evaluates to this condition. The default is to *include*.

Syntax:

```
Filter 'filter-list-name' Config>SET-ACTION ?
INCLUDE
EXCLUDE
TAG
```

a) SET-ACTION INCLUDE

Example:

```
Filter 'sample' Config>SET-ACTION INCLUDE
Filter 'sample' Config>
```

b) SET-ACTION EXCLUDE

Example:

```
Filter 'sample' Config>SET-ACTION EXCLUDE
Filter 'sample' Config>
```

c) SET-ACTION TAG

Example:

```
Filter 'sample' Config>SET-ACTION TAG
Enter a tag value[1]? 3
Filter 'sample' Config>
```

4.7. EXIT

Use the exit command to return to the previous prompt.

Syntax:

```
Filter 'filter-list-name' Config>EXIT
```

Example:

```
Filter 'sample' Config>EXIT
Filter Config>
```

Chapter 14
NetBIOS Filtering and Caching commands

1. About Threading

Threading is the process whereby the network protocol (IPX, DNA, IP, AppleTalk and Apollo) of the Token Ring end station discovers a route over segments of an SRB Network.

Threading is no different from the SRB operation. It is how threading is implemented by the end station that is different. The following sections describe threading for IP, DNA, IPX, AppleTalk and Apollo.

2. IP Threading with ARP

IP end stations use Address Resolution Protocol (ARP) REQUEST and REPLY packets to discover an RIF. Both IP end stations and the bridges participate in the route discovery and forwarding process. The following steps describe IP threading.

1. An IP end station maintains an ARP table and an RIF table. It uses the MAC address in the ARP table as a cross reference for the destination RIF in the RIF table. If an RIF does not exist for that specific MAC address, the end station transmits an ARP REQUEST packet with an ARE (All Routes Explore) or an STE (Spanning Tree Explore) onto the local segment.
2. All bridges on the local segment capture the ARP REQUEST packet and send it over their connected networks.
3. As the ARP REQUEST packet continues its search for the destination end station, each bridge that forwards it adds its own bridge number and segment number to the RIF in the packet. As the frame continues to pass through the bridged network, the RIF complies a list of bridge and segment number pairs describing the path to the destination.
4. When the ARP REQUEST packet finally reaches its destination, it contains the exact sequence of bridge and segment numbers from source to destination.
5. When the destination end station receives the frame, it puts the MAC address and its RIF into its own ARP and RIF tables. If the destination end station receives any other ARP REQUEST packets from the same source, it drops that packet.
6. The destination end station then generates an ARP REPLY packet including the RIF and sends it back to the source end station.
7. The source end station receives the learned-route path. It puts the MAC address and its RIF into the ARP and RIF tables. The RIF is then attached to the data packet and forwarded onto the destination.
8. Aging of RIF entries is handled by the IP ARP refresh timer.

3. DNA Threading

Digital Network Architecture (DNA) end stations use ARE (All Routes Explore) to discover a route. Both the DNA end stations and the bridges participate in the route discovery process and forwarding. The following steps describe the DNA threading process.

1. If there is no entry in the RIF table for the MAC address, an entry is created with the state *NO_ROUTE*. When this occurs the end station sends the data packet out with an STE attached. The STE is used for discovery without attempting to flood the network with ARE.
2. The end station then transmits an ARE in a loop-back frame to the destination MAC address.
3. All bridges on the local segment capture the STE and loop-back frame and send it over their connected networks.
4. As the packets continue their search for the destination end station, each bridge that forwards it adds its own bridge number and segment number to the RIF in the STE and the ARE. As the frames passes through the bridged network, the RIF complies a list of bridge and segment number pairs describing the path to the destination.
5. When the STE and loop-back frame finally reaches the destination, it contains the exact sequence of bridge and segment numbers from source to destination.
6. When the destination end station receives the loop-back frame it puts the MAC address and the RIF of the source station into its own RIF table. If an RIF already exists for that entry, it either updates the RIF if that previous entry is an *ST_ROUTE* or it ignores the RIF. In any case the entry state is changed to *HAVE_ROUTE.*
7. The destination end station then sends the loop-back reply frame including the specific RIF back to the source end station.
8. The source end station receives the learned specific route path. It puts the RIF into the RIF table and the entry changes to *HAVE_ROUTE*.
9. Packets destined for a functional address are sent with an STE. DNA end stations can create an RIF entry using this STE frame. When this happens the state of the entry is changed to *ST_ROUTE*.

The DNA end stations contain an independent RIF timer. When this timer expires for a specific RIF entry, an ARE in a loop-back packet is sent out to that specific destination. When the loop-back frame returns, the RIF entry is updated. If the destination end station is on the same ring and the loop-back frame contains no RIF, the loop-back packet is returned with no RIF entry.

4. Apollo Threading

Apollo end stations use STE frames to discover a route. Both the Apollo end stations and the bridges participate in the route discovery process and forwarding. The following steps describe the Apollo threading process.

1. If there is no entry in the RIF table for the MAC address the data packet is sent out with an STE. An entry is added to the RIF table designated as NO_ROUTE.
2. The end station then transmits another STE with XID for the destination MAC address.
3. All bridges on the local segment capture the STE and send it over their connected networks.
4. As the packets continue their search for he destination end station, each bridge that forwards it adds its own bridge number and segment number to the RIF in the STE. As the frames passes through the bridged network, the RIF complies a list of bridge and segment number pairs describing the path to the destination.
5. When the STEs finally reach the destination, it contains the exact sequence of bridge and segment numbers from source to destination.
6. When the destination end station receives the STE with XID, it puts the MAC address and the RIF of the source station into its own RIF table. If an RIF already exists for that entry, it either updates the RIF if that previous entry is an *ST_ROUTE* or it ignores the RIF. In any case the entry state is changed to *HAVE_ROUTE*.
7. The destination end station then sends an XID reply frame including the specific RIF back to the source end station.
8. The source end station receives the learned specific route path. It puts the RIF into the RIF table and the entry changes to *HAVE_ROUTE*.
9. Packets destined for a functional address are sent with an STE with no XID. Apollo end stations can create an RIF entry using this STE frame. When this happens the state of the entry is changed to *ST_ROUTE*.

The Apollo end stations contain an independent RIF timer. When this timer expires for a specific RIF entry, an STE with XID packet is sent out to that specific destination. When the XID reply frame returns, the RIF entry is updated. If the destination end station is on the same ring, the loop-back packet is sent and returned with no RIF entry.

5. IPX Threading

IPX end stations check each packet they receive for an RIF. If the RIF does not exist in the table, they add the RIF to the table and designate that route as HAVE_ROUTE. If the RIF indicates that the packet came from an end station on the local ring, the route is designated as ON_RING.

If the end station needs to send out a packet and there is no entry in the RIF table for the MAC address, the end station transmits the data as an STE.

When the RIF timer expires, the entry in the table is cleared and won’t be reentered until another packet arrives containing an RIF for that entry.

6. Threading AppleTalk 1 and 2

AppleTalk end stations use ARP and XID packets to discover a route. Both the AppleTalk end stations and the bridges participate in the route discovery process and forwarding. The following steps describe the AppleTalk threading process.

1. If an RIF does not exist for a specific MAC address, the end station transmits an ARP REQUEST packet with an ARE (All Routes Explore) onto the local segment.
2. All bridges on the local segment capture the ARP REQUEST packet and send it over their connected networks.
3. As the ARP REQUEST packet continues its search for the destination end station, each bridge that forwards it adds its own bridge number and segment number to the RIF in the packet. As the frames passes through the bridged network, the RIF complies a list of bridge and segment number pairs describing the path to the destination.
4. When the destination end station receives the frame, it puts the MAC address and its RIF into its own ARP and RIF tables and the state of the entry is designated as *HAVE_ROUTE*. If the destination end station receives any other ARP REQUEST packets from the same source, it drops that packet.
5. The destination end station then generates an ARP REPLY packet including the RIF and sends it back to the source end station with the direction bit in the RIF flipped.
6. The source end station receives the learned route path. The MAC address and its RIF are then entered into the ARP and RIF tables and the state designated as *HAVE_ROUTE*. If the RIF indicates that the packet came from an end station on the local ring, the route is designated as *ON_RING*.
7. If the RIF timer expires, an XID is sent out with an RE and the state is changed to *DISCOVERING*. If no XID reply is received, the entry is discarded.