Bridge

bintec Dm717-I

Legal Notice

Warranty

This publication is subject to change.

bintec offers no warranty whatsoever for information contained in this manual.

bintec is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

I Related Documents

bintec Dm715-I Bandwidth Reservation System

bintec Dm716-I DLSw Protocol

bintec Dm751-I VLAN

Chapter 1 Fundamentals of Bridging

1.1 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.

Bridges are used 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.

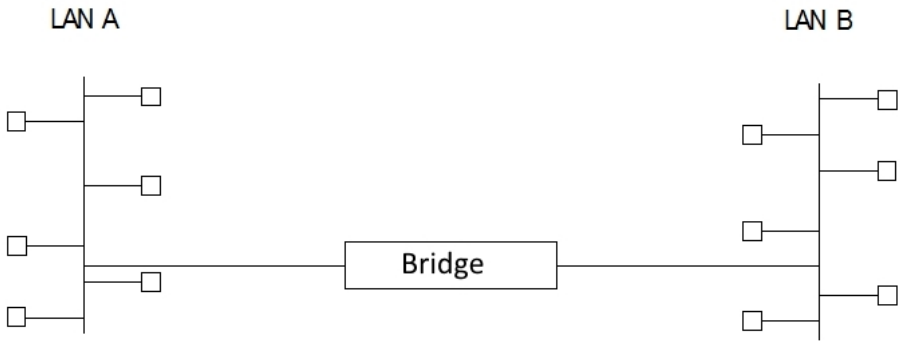

Fig. 1: **Simple Bridge Connecting Two Homogeneous Ethernet LANs**

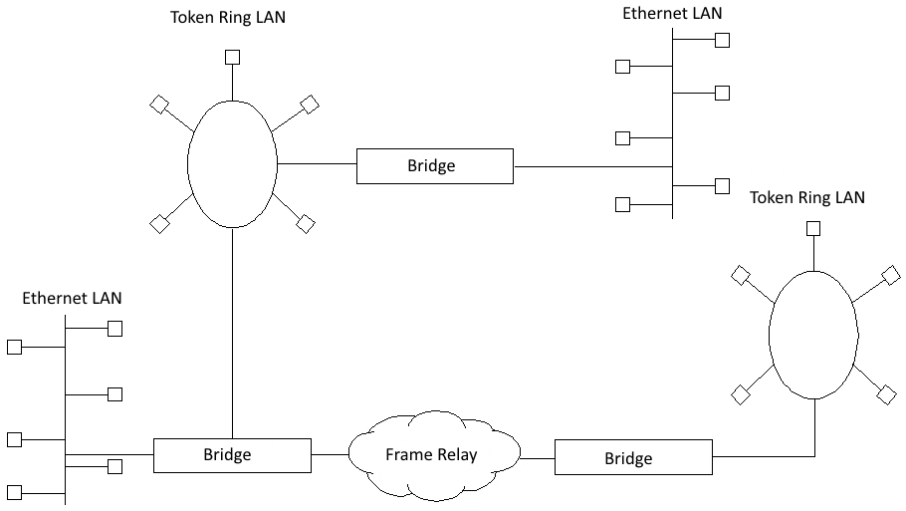

Fig. 2: **Homogeneous and Heterogeneous Bridging Configurations**

1.2 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 the *OSI* model layer 3 (network layer) while the bridges connect LANs at layer 2 (data link layer).

1.2.1 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 the 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.

1.2.2 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 main 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.

1.2.3 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 more 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.

1.2.4 Bridging interfaces

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

- Ethernet.
- Token Ring.
- Frame Relay.
- PPP.
- ATM.
- Tunnel IP.

Ethernet interfaces support transparent bridging.

The Token Ring interface supports source routing and transparent bridging.

The rest of the interfaces provide point-to-point connectivity for transparent and source routing traffic. It is important to note that a bridge configuration over an interface of this type 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 interface 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.

1.3 Bridge 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 where 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 where 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 bintec'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.

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

1.4 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.

1.4.1 Example 1: Local bridge connecting two LANs

Fig. 3 on page 4 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.

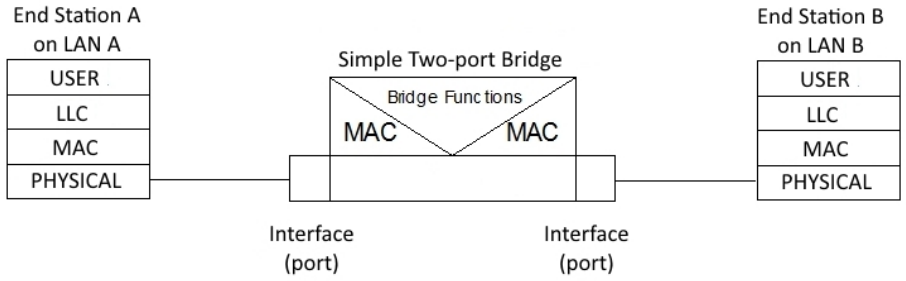

Fig. 3: **Two-port Bridge Connecting Two LANs**

1.4.2 Example 2: Remote bridging over a serial link

Fig. 4 on page 4 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.

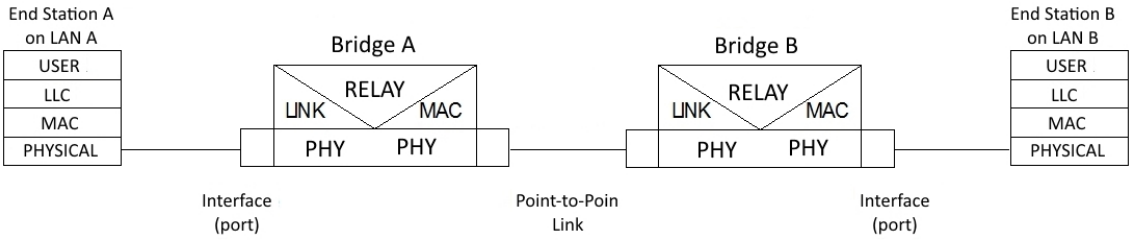

Fig. 4: **Bridging Over a Point-to-Point Link**

Fig. 5 on page 5 illustrates the encapsulation process.

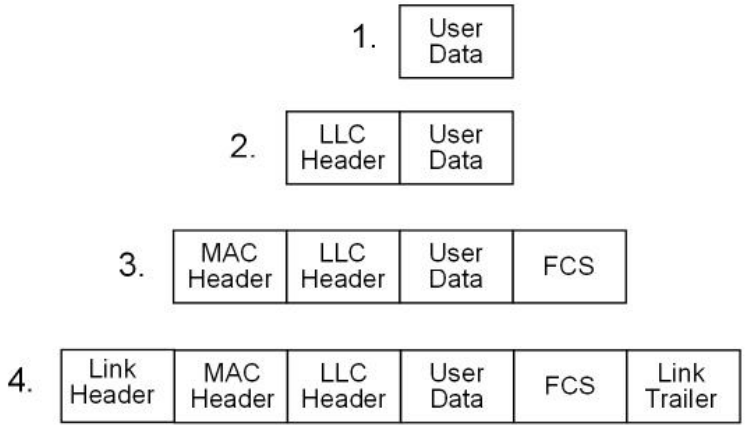

Fig. 5: **Data Encapsulation over a Point-to-Point Link**

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 a 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.

1.4.3 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.

Fig. 6 on page 6 shows the CSMA/CD and Token Ring MAC frame formats supported by the bridges. The specific frames are detailed in the following section.

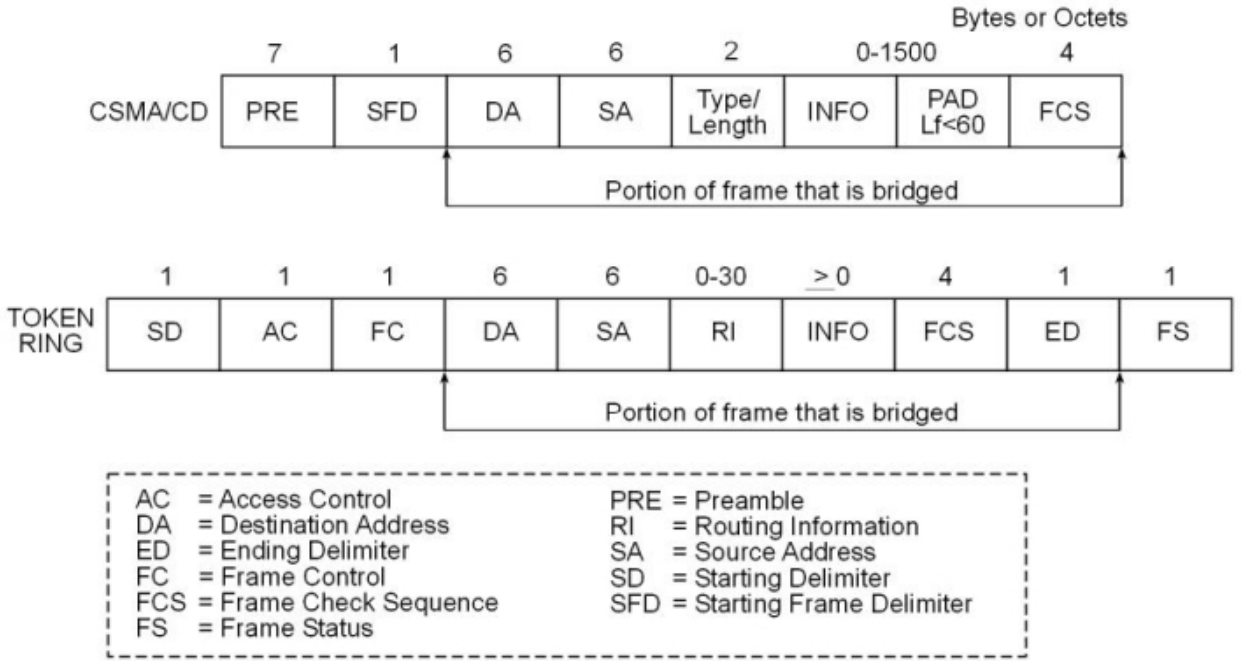

Fig. 6: **MAC Frame Format Samples**

1.4.4 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) or a broadcast address (all stations as 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 can never be multicast or broadcast.
- *Type/Length.* If the value of this field is less than 0x0600, it is interpreted as length and specifies the length, in bytes, present in the MAC frame INFO field. These are normally known as IEEE 802.3 frames. If the value of this field is greater than 0x0600, then it is interpreted as a higher layer protocol encapsulated in the MAC frame. This is known as Ethernet-II frame.
- *Info (INFO).* Data present in the MAC frame.
- *Pad.* Sequence of bytes that ensures 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.

1.4.5 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).* Specifies the device the frame is addressed to. Same as CSMA/CD, except that bit format is non-canonical.
- *Source Address (SA).* Identifies the specific station that originates the frame.

- *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).* Data present in the MAC frame.
- *Frame Check Sequence (FCS). A* 32-bit cyclic redundancy check value. This value is based on all fields, starting with the destination address.
- *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 than the final one of a multiple *frame* transmission.
- *Frame Status (FS).* Contains the address recognized (A) and frame copied (C) bits.

Chapter 2 Using Transparent Bridging (STB)

2.1 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.

2.2 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.

2.3 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:

Versions before 11.01.00:

- **protocol asrt**. Accesses the ASRT bridge configuration menu.

- **bridge**. Enables transparent bridging.
- **port** *port#*. Enables bridging for a determined interface. Execute this command for all LAN/WAN interfaces over which the bridge is going to operate.

Versions from 11.01.00:

- **add device bv1***0*. Enables transparent bridging for main bridge.
- **protocol asrt**. Accesses the ASRT bridge configuration menu.
- **port** *port#*. Enables bridging for a determined interface. Execute this command for all LAN/WAN interfaces over which the bridge is going to operate.

After completing the procedures just described, run **list bridge** to check the configuration.

To make changes to the configuration, see *Bridging Configuration* on page 32 of this manual. After making the changes to the configuration, restart the router so the new configuration activates.

2.4 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:

- 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. The other bridges select a port as a root port. This port is the least cost port associated with reach the root bridge.
- 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.
- The non-designated bridges on the LANs put each port that has not been selected as a root port into a *blocked* state. In a 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.

2.5 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 *Fig. 7* on page 10 explains how to do this.

Fig. 7 on page 10 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

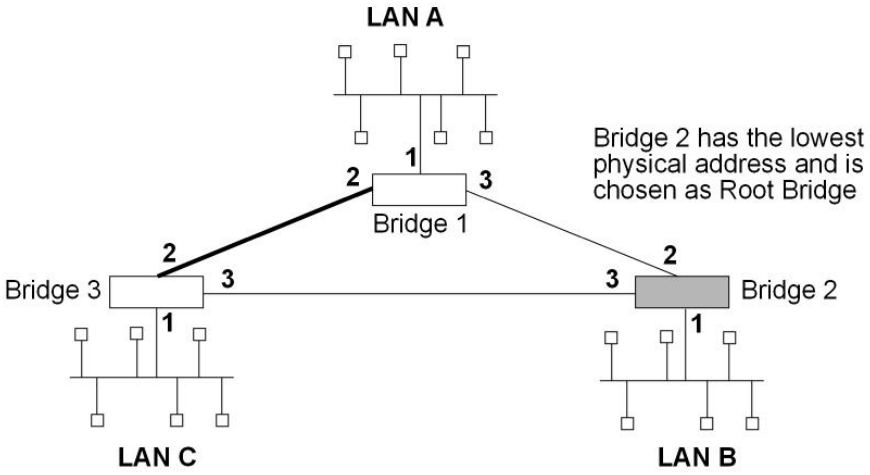

Fig. 7: **Networked LANs before Spanning Tree**

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 *Fig. 8* on page 10 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.

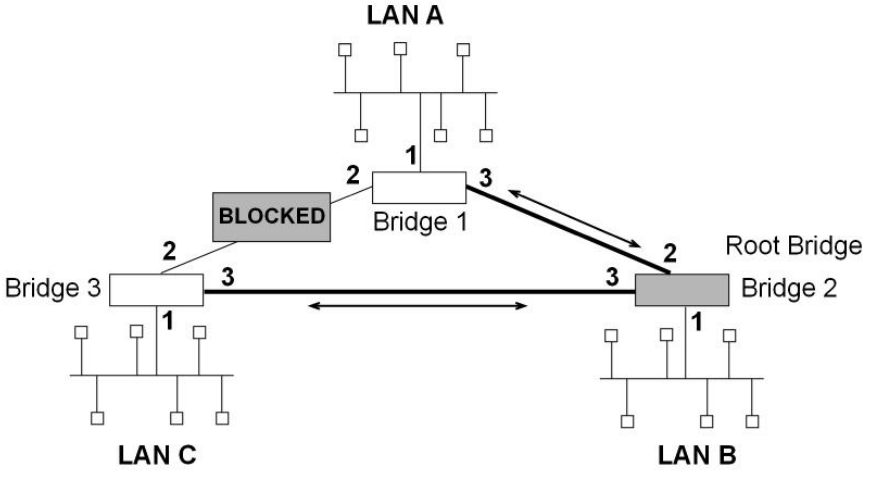

Fig. 8: **Spanning Tree Created with Default Values**

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 *Fig. 9* on page 11 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.

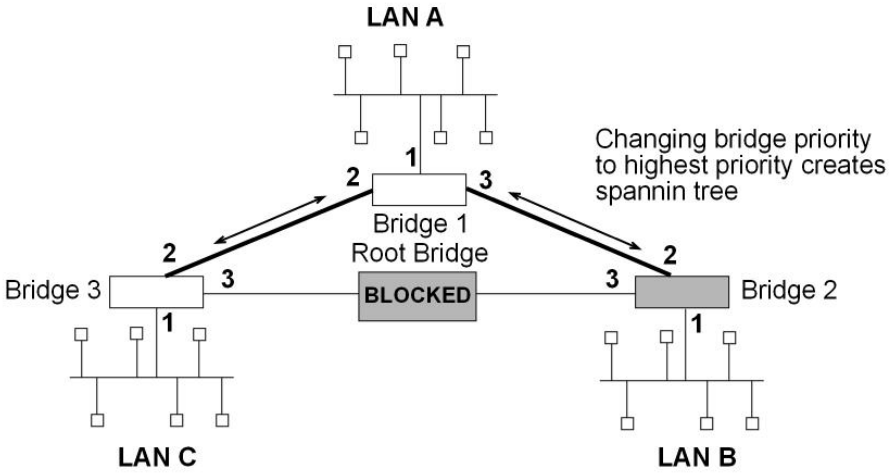

Fig. 9: **User-adjusted Spanning Tree**

2.6 Spanning tree bridges and Ethernet packet format translation

The SSTB protocol forwards packets complying 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)

3.1 About SRB

Source Route Bridging (SRB) is a method of forwarding frames through a bridged network where the source station identifies the route the frame will follow. In a distributed routing scheme, routing tables at each bridge determine the path 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 . *Fig. 10* on page 12 It can also connect remote LANs through a telecommunications link operating at speeds up to E1.

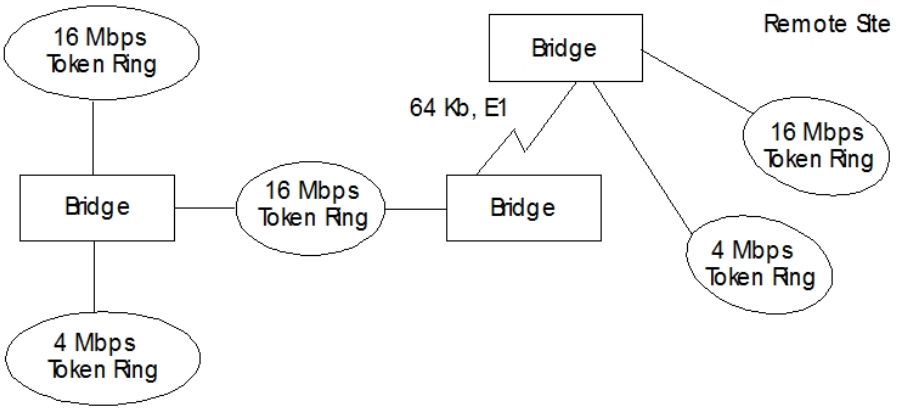

Fig. 10: **Source Routing Bridge Connectivity Sample**

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. 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).
- *FCS preservation.* bintec bridges preserve Frame Check Sequence of the Specifically Routed Frames (SRF). This protects against data corruption of the bridged frames.

3.2 Enabling SRB

The following information outlines the initial steps required to enable the SRB bridging option.

Versions before 11.01.00:

• **protocol asrt**. Accesses ASRT bridge configuration menu.

• **bridge**. Enables transparent bridging.

• **port** *port#.* Enable bridging for a particular interface. Execute this command for all LAN/WAN interfaces over which the bridge is going to operate.

Versions from 11.01.00:

- **add device bvi** *0*. Enables bridging for main bridge.
- **protocol asrt**. Access the ASRT bridge configuration menu.
- **port** *port#*. Enables bridging for a particular interface. Execute this command for all LAN/WAN interfaces over which the bridge is going to operate.
- **no transparent** *port#.* Disables transparent bridging on a bridge port.
- **source-routing** *port# segment#.* Enables bridge source-routing on a port.

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

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, enter **list bridge** to verify your configuration.

If you want to make changes to the configuration, see *Bridging Configuration* on page 32 of this guide. After you finish changing the configuration, restart the router for the new configuration to take effect.

3.3 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.

3.4 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. *Fig. 11* on page 13 provides a closer look at the format of the source address field of a source routing frame.

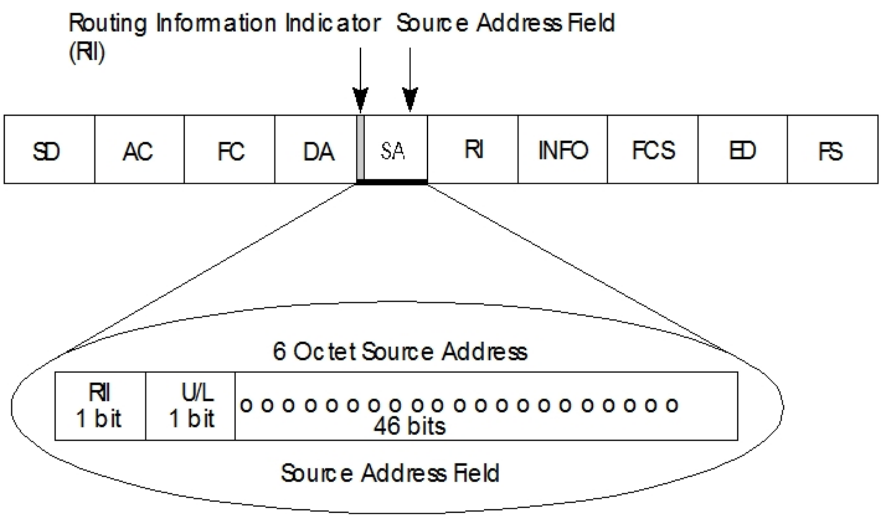

Fig. 11: **802.5 Source Address Format**

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. *Fig. 12* on page 14 provides a closer look at the format of the Routing Information Field.

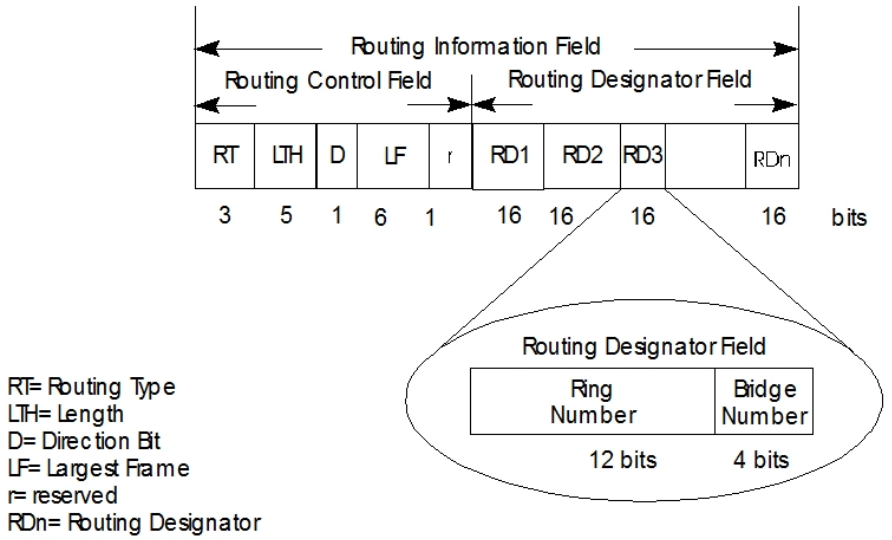

Fig. 12: **802.5 Routing Information Field**

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 frame type is used to find a remote station. 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 contain a specific destination address. During route discovery phase, this type of frame is used as a response to an 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 they are specified in the routing information field in (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).

3.5 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. This option lets you **simulate** a spanning tree network (do not confuse this with the spanning tree protocol).

3.5.1 Simulating a spanning tree network

SRB bridges can 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 *Fig. 13* on page 15, the 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.

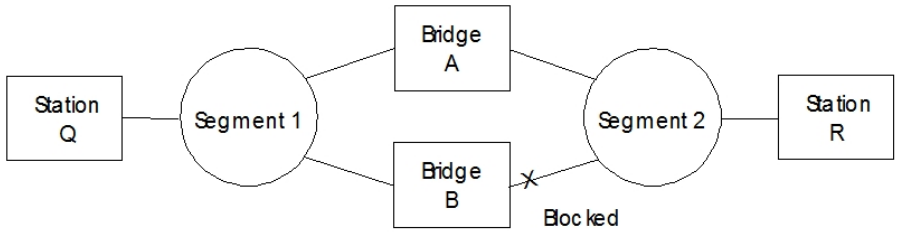

Fig. 13: **Sample Parallel Bridge**

3.6 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)

4.1 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.

4.2 Enabling SR-TB

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

Versions before 11.01.00:

- **protocol asrt**. Accesses the ASRT bridge configuration menu.
- **bridge**. Enables bridging.
- **port** *port#*. Enables bridging for a particular interface. Execute this command for all LAN/WAN interfaces over which the bridge is going to operate.
- **no transparent** *port#.* Disables transparent bridging on a bridge port.
- **source-routing** *port# segment#*. Enables bridge source-routing on a port.
- **sr-tb-conversion** *segment# mtu#.* 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.

Versions from 11.01.00:

- **add device bvi** *0*. Enables bridging for main bridge.
- **protocol asrt**. Accesses the ASRT bridge configuration menu.
- **port** *port#*. Enables bridging for a particular interface. Execute this command for all LAN/WAN interfaces over which the bridge is going to operate.
- **no transparent** *port#*. Disables transparent bridging on a bridge port.
- **source-routing** *port# segment#*. Enables bridge source-routing on a port.
- **sr-tb-conversion** *segment# mtu#.* 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, 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 *Bridging Configuration* on page 32 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.

4.3 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.

Fig. 14 on page 17 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.

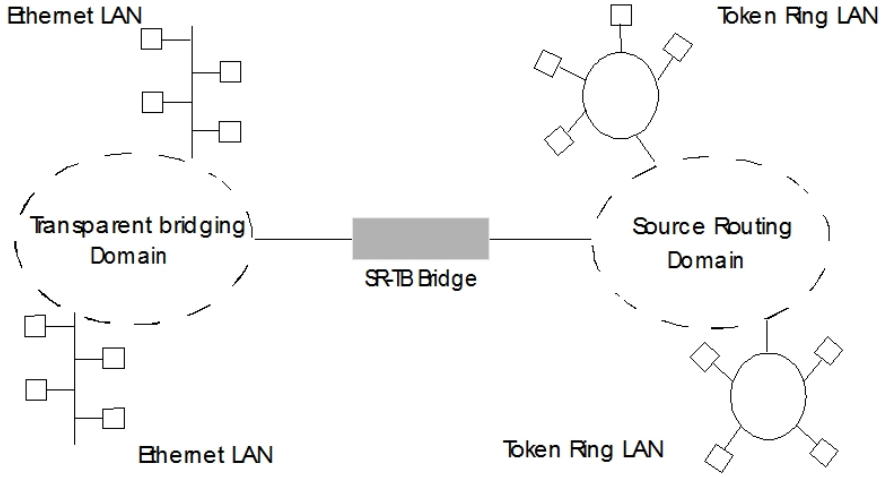

Fig. 14: **SR-TB Bridge Connecting Two Domains**

4.3.1 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.

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 *SR-TB Bridge Decision Table* on page 17.

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 the 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, it 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 RII 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.

4.3.2 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. *Fig. 15* on page 19 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.

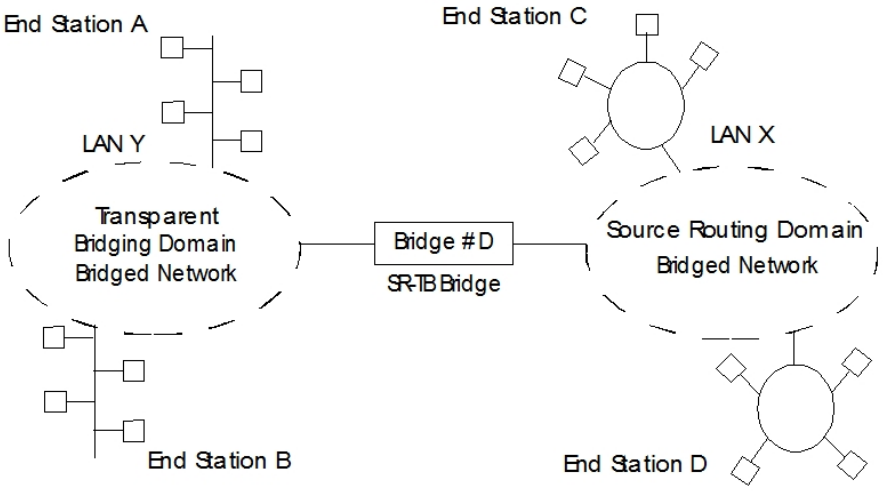

Fig. 15: **SR-TB Bridging Examples**

4.3.2.1 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.

4.3.2.2 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.

4.3.2.3 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 RII 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.

4.3.2.4 Example 4: Frame sent from end station C to end station A

If the frame is sent non-broadcast, SR-TB scans the RII 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.

4.4 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

5.1 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.

Route/Bridge Decision Table on page 21 shows how the destination address field determines the question of whether to “Bridge or Route”.

Route/Bridge Decision Table

If destination MAC Addressing contains:	**Action the Bridge Takes**
Internet Address	Passes the frame to the configured protocol that routes the frame.
Multicast or Broadcast Address	If the frame belongs to a configured protocol, it is passed to the protocol forwarder to be routed. Frame bridging is executed if the bridge is enabled for the protocol. If the frame does not belong to a configured protocol, it bridges the frame.
Other Unicast	If the frame belongs to a configured protocol, it discards the frame or frame bridging is executed if the bridge is enabled for the protocol. If the frame does not belong to a configured protocol, it bridges the frame.

5.2 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 behavior and generates the correct frame format. 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.

5.3 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.

5.4 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.

5.4.1 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 the 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.

5.4.2 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 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 SR 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.

5.5 Processing BPDUs

This section describes the configurable characteristics in the bridge to prevent a port from sending or receiving spanning tree frames (BPDUs: Bridge Protocol Data Units).

5.5.1 Filtering BPDUs

BPDUs filtering can be configured globally or per port.

- If this is configured in a port by running **set spanning-port <port> bpdu-filter enable**, the port doesn't transmit or receive BPDUs.

Note

Filtering BPDUs in a port is equivalent to disabling the spanning tree in the port and could provoke loops establishing in the network.

- If this is globally configured by running **set spanning-tree bpdu-filter default**, it is enabled in all the ports, which, as they don't have a specific BPDU filtering configuration associated, behave as *edge ports* i.e., ports directly connected to a station. The *rapid spanning tree* states machine detects whether a port is an *edge port* or not. If BPDU filtering is enabled globally, a port in an *edge* state doesn't transmit BPDUs. When the port receives a BPDU it passes to a *no edge* state and stops BPDU filtering. From that point the port can transmit and receive BPDUs.

If a port has BPDU filtering enabled or disabled, it ignores the global configuration. When the BPDU filtering configuration for a port is not specified (default configuration), the global configuration and the *EdgePort* state variable is used to determine if BPDU filtering should be executed or not. The following table describes the BPDU filtering operation in a port depending on the configuration.

Configuration per port	Global configuration	EdgePort operating value	BPDU filtering
Disable	-	-	*Disable*
Enable	-	-	*Enable*
Default	Disable	-	*Disable*
Default	Enable	EdgePort	*Enable (If the port receives a BPDU, it passes to a Non Edge Port and BPDU filtering is disabled)*

Default	Enable	Non EdgePort	*Disable*

5.5.2 BPDU guard

The BPDU Guard is a mechanism that protects against topology changes provoked by spanning tree operations. This function is independent of BPDU filtering. When both are configured in a port, the BPDU Guard has priority over BPDU filtering.

If BPDU guard is configured in a port and a BPDU is received, this is considered an error situation and forces the port to pass to a disabled state due to error detection. For further information on this state and how to recoup an interface in this state, please see manual bintec *Dm794-I Interface*.

In the same way as BPDU filtering, the BPDU guard can be configured globally or per port.

- If this is configured in a port by running **set spanning-tree <port> bpdu-guard enable**, the port cannot receive BPDUs. If it receives a BPDU, the port passes to a disabled (due to error state) and stops transmitting and receiving frames.
- If this is globally configured by running **set spanning-tree bpdu-guard default**, it is enabled in all the ports, which, as they don't have a specific BPDU guard configuration associated, behave as *edge ports* i.e., ports directly connected to a station. The *rapid spanning tree* states machine detects whether a port is an *edge port* or not. If BPDU guard is enabled globally and a port in an *edge* state receives a BPDU, the port passes to a disabled state due to error and stops sending and transmitting frames.

If a port has BPDU guard enabled or disabled, it ignores the global configuration. When the BPDU guard configuration for a port is not specified (default configuration), the global configuration and the *EdgePort* state variable is used to determine if BPDU guard should be executed or not. The following table describes the BPDU guard operation in a port, depending on the configuration.

Configuration per port	**Global configuration**	**EdgePort operating value**	**BPDU Guard**
Disable	-	-	*Disable*
Enable	-	-	*Enable*
Default	Disable	-	*Disable*
Default	Enable	EdgePort	*Enable*
Default	Enable	Non EdgePort	*Disable*

Chapter 6 Using IP Tunneling

6.1 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 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. *Fig. 16* on page 24 shows an example of an IP internetwork using the tunnel feature in its configuration.

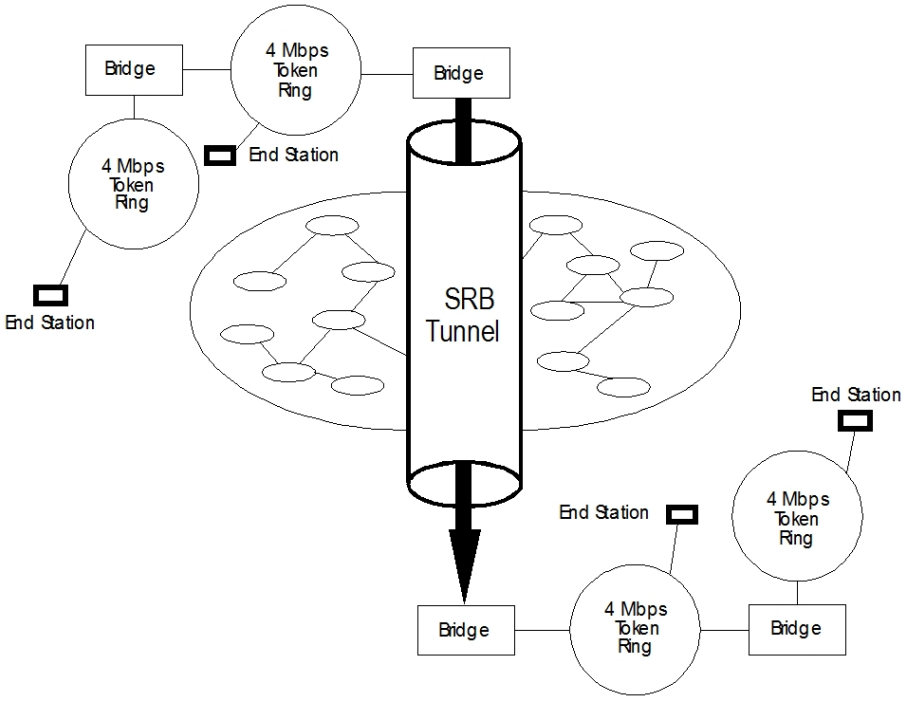

Fig. 16: **End Stations see Routing Across Complex IP Internet as One Hop**

The bridges participating in tunneling treat the IP Internet as one of the bridge segments. 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.

6.1.1 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 up-

dates to the best path. This rerouting is completely transparent to the end stations.

Chapter 7 Multiple Bridge Entities

7.1 What is a bridge instance?

A bridge instance can, to all effects, be considered as an independent bridge. The bintec devices allow you to configure various bridge instances so a single device is equivalent to various bridges. Each instance uses independent configuration parameters and independently executes the Spanning Tree algorithm.

Each virtual bridge instance is assigned certain specific interfaces or ports over which the bridge is executed. An interface cannot form part of various bridge instances.

Each virtual bridge instance is identified with a name. The virtual bridge instance identified with 0 is known as the *main bridge*. You can define up to a total of eight virtual bridge instances in a device.

A new interface is automatically created in the device for each virtual bridge instance when the virtual bridge is enabled. From version 11.01.00, the interface has to be manually created, which automatically enables the virtual bridge. This is known as a BVI (*Bridge Virtual Interface*). This represents the group of interfaces included in the bridge. BVI interfaces cannot be added as bridge ports.

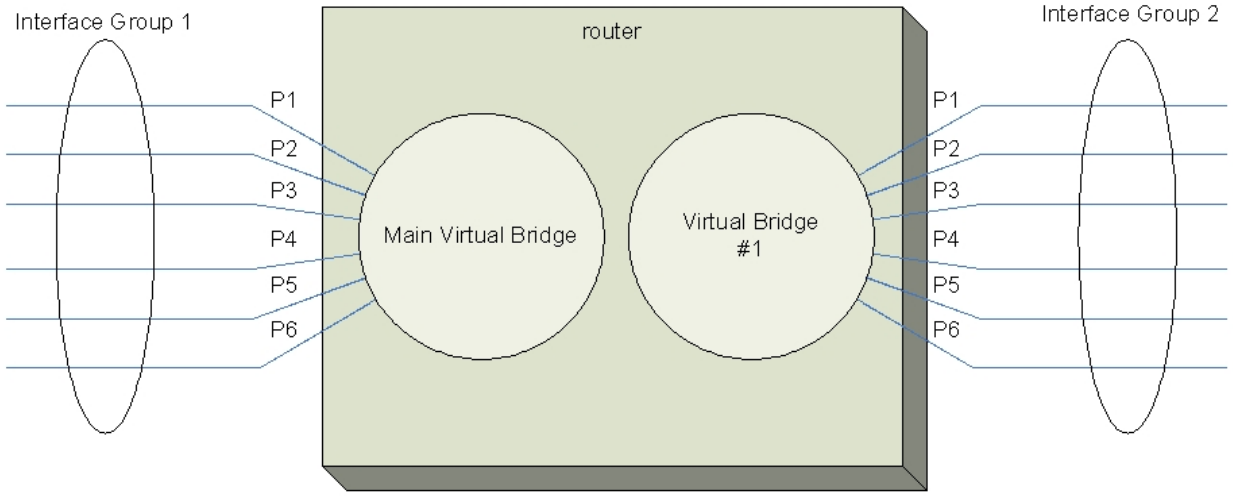

Fig. 17: **Diagram of a device with two virtual bridges**

7.2 Considerations

There are a number of considerations to take into account when configuring various bridge instances:

- The same interface cannot belong to more than one bridge.
- A virtual bridge cannot exchange traffic with other virtual bridges.
- DLSw traffic can only travel over the main virtual bridge.
- BAN traffic can only travel over the main virtual bridge.
- BVI interfaces cannot be added as bridge ports.
- If an interface belongs to a bridge, an IP address is assigned to the latter, its associated bridge is disabled and the traffic between the interfaces can only be routed.

Chapter 8 Integrated Routing and Bridging

8.1 Integrated routing and bridging

Integrated routing and bridging (IRB) allows a single device to behave as a bridge for some packets and as a router for others. Currently it's only possible to execute integration over IPv4 (and ARP) and IPv6 packets.

IRB is disabled by default. The **irb** command enables *integrated routing and bridging* in a specific bridge instance. From version 11.01.00, **irb** is no longer necessary: this feature is automatically activated when a routing protocol is configured by running **route-protocol**.

A bridge configured without IRB executes bridging on received packets, but does not forward them. This behavior changes when an IPv4 or IPv6 address is configured over some of the interfaces participating in the bridge, so the protocol packets said configured address belongs to (IPv4 and/or IPv6) are routed but not bridged.

If you configure **irb**, the processing on each packet corresponding to a particular protocol depends on the bridge configuration for said protocol. By default, a protocol is configured to be bridged and not routed. By running **route-protocol**, routing can be activated for a particular protocol over the bridge.

Additionally, bear in mind that it's possible to define filters that affect the protocol. Configuration for said protocols can exclude a protocol both from the routing as well as from bridging, so the packets corresponding to said protocol, which reach the router through one of the bridge interfaces, are simply dropped.

8.2 Bridge virtual interface (BVI)

The integrated functionality of bridging and routing is based on the bridge virtual interface concept (BVI).

A BVI interface is an additional interface in the router representing a group of interfaces included in a bridge. You can assign IPv4 and/or IPv6 addresses to the BVI interface (if the bridge has IRB enabled) and can be used in the configuration for any of the protocols in the IPv4 and IPv6 group of protocols.

A BVI interface cannot form part of any bridge port.

Each bridge is associated with a BVI interface, which is automatically created on enabling the corresponding virtual bridge. The bvi0 interface corresponds to the main bridge instance, the bvi1 interface to the bridge virtual 1 entity and so on. From version 11.01.00 the BVI interface has to be manually created (**add device bvi <id>**), and the associated bridge is automatically enabled.

Fig. 18 on page 28 shows a diagram that represents a group of decisions taken when a packet from a particular protocol is received over one of the interfaces pertaining to a bridge.

For a packet to be delivered to the corresponding protocol forwarder, said packet must be routable. There are three things to check this condition:

a) The packet has a broadcast destination MAC address.

b) The packet has a multicast destination MAC address.

c) The packet has a destination MAC address that the bridge has registered as belonging to the device itself.

The key to bridging and routing integration lies in the fact that the BVI has appropriate addresses at both layer 2 (MAC address) and layer 3 (in this case, IP address).

The MAC address for a BVI is established in the following way:

(1) If an address has been assigned to the bridge through configuration (**set bridge**), the BVI takes this as its own.

(2) If the bridge includes at least one interface with its own MAC address (Ethernet or Token Ring), the BVI uses this as its own from the moment said interface becomes operative. If there are various interfaces with MAC addresses, one of these is selected.

(3) If there is no interface with a MAC address, then one of the router's preassigned MAC addresses is assigned to the BVI.

(4) If the device does not have a free preassigned MAC (because all the MACs it has have been assigned to other interfaces), the BVI interface remains down. In this case you can assign a locally administrated MAC, configuring it in the bridge through **set bridge mac-address**. This allows the BVI interface to activate (up) and operate normally.

Said MAC address remains registered in the bridging tables as own address, so subsequently packets destined to this address can be identified as routable.

When a packet is routed through a BVI for fowarding, the bridge selects the output interface based on the packet destination MAC address. This address must appear in the registered MAC addressing table for one of the bridge ports: it must be the result of a previous ARP.

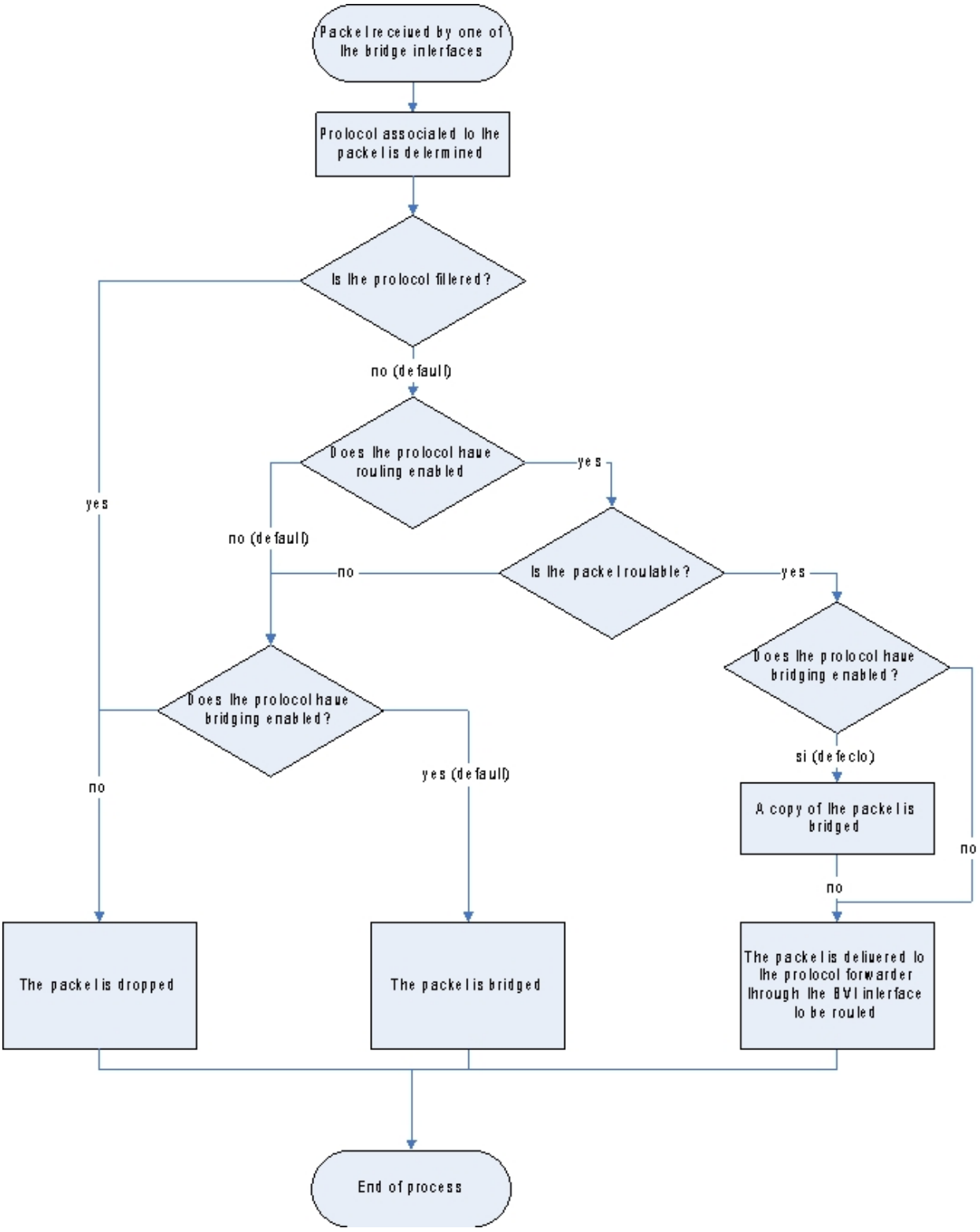

Fig. 18: **Flow chart showing the processing of a packet with IRB enabled**

8.3 Enabling integrated routing and bridging

The **irb** command (bridge configuration menu) enables the integrated routing and bridging feature for said bridge. From version 11.01.00, this command is no longer needed since this feature is activated when a routing protocol is configured. By default, the bridge maintains compatibility with previous configurations. If **irb** is not enabled, the following is fulfilled:

(1) If none of the bridge participating interfaces has IP (IPv4 or IPv6) addresses, the IP with no addresses (IPv4 or IPv6) can execute bridging but not routing.

(2) If one of the bridge participating interfaces has an IP (IPv4 or IPv6) address, the IP that has addresses (IPv4 or IPv6) can execute routing but not bridging.

(3) IP (IPv4 or IPv6) addresses cannot be added to the BVI associated with the bridge.

When integrated bridging and routing is enabled, the following is implemented:

(1) Protocol routing is disabled by default (this can be enabled by running **route-protocol**). This does not apply to versions from 11.01.00, since integrated bridging and routing is enabled only when a routing protocol is configured.

(2) Protocol bridging is enabled by default (this can be disabled by running **no bridge-protocol**).

(3) IP (IPv4 or IPv6) addresses can be added to the BVI associated with the bridge.

8.4 Enabling protocol routing

Run **route-protocol <protocolname>** (bridge instance configuration menu) to enable the routing protocol. Default is the bridge does not route routable packets for a protocol unless specifically enabled through **route-protocol**.

From version 11.01.00, integrated routing and bridging is enabled automatically when a protocol is configured using this command.

Disable protocol routing by running **no route-protocol<protocolname>**.

The only protocols that are currently configurable are IPv4 and IPv6.

8.5 Disabling protocol bridging

The **no bridge-protocol <protocolname>** command disables protocol bridging. Therefore packets belonging to said protocol are only routed where they are routable (but never bridged).

Enable protocol bridging by running **bridge-protocol <protocolname>**. Default is all protocols are bridged.

The only currently configurable protocols are IPv4 and IPv6.

8.6 IRB with bandwidth reservation

Bandwidth Reservation (BRS) can be enabled in the bridging ports to apply quality of service policies (QoS). These policies are applied to bridge frames when going out through the ports where BRS is enabled.

The method to classify traffic in bridging ports is based on layer 2 policies such as the MAC filtering feature. However, BVI can be configured so routed traffic is classified before being forwarded through the bridge. Layer 3 policies (e.g., IP addresses of the IP header ToS field) can then be applied to packets routed through the bridging port where Bandwidth Reservation is enabled.

Enable classification in BVI by running **qos-pre-classify**.

Disable classification in BVI by running **no qos-pre-classify**.

8.7 BVI subinterfaces

Creates BVI subinterfaces associated with a BVI interface. Different encapsulated VLANs can be assigned to each BVI subinterface. Consequently, the different encapsulated VLANs for different IP networks can be defined.

To create a BVI subinterface, run **add device** from the general configuration menu.

Syntax:

```
Config>add device bvi-subinterface <BVI base interface> <subinterface number>
Config>
```

Example:

```
Config>add device bvi-subinterface bvi0 1
Config>
```

Once created, access the subinterface configuration by running **network**.

Example:

```
Config>net bvi0.1
```

```
-- BVI Subinterface Configuration --
bvi0.1 config>
```

From the BVI subinterface configuration menu, configure the encapsulated VLAN to use by entering **encapsulation dot1q**.

Example:

```
bvi0.1 config>encapsulation dot1q 101
```

8.8 IRB with bridge spoofing

The Bridge Spoofing feature allows transparent backup in routing scenarios through layer 2 spoofing techniques.

The aim of Bridge Spoofing is to add a backup service in a scenario similar to the following figure:

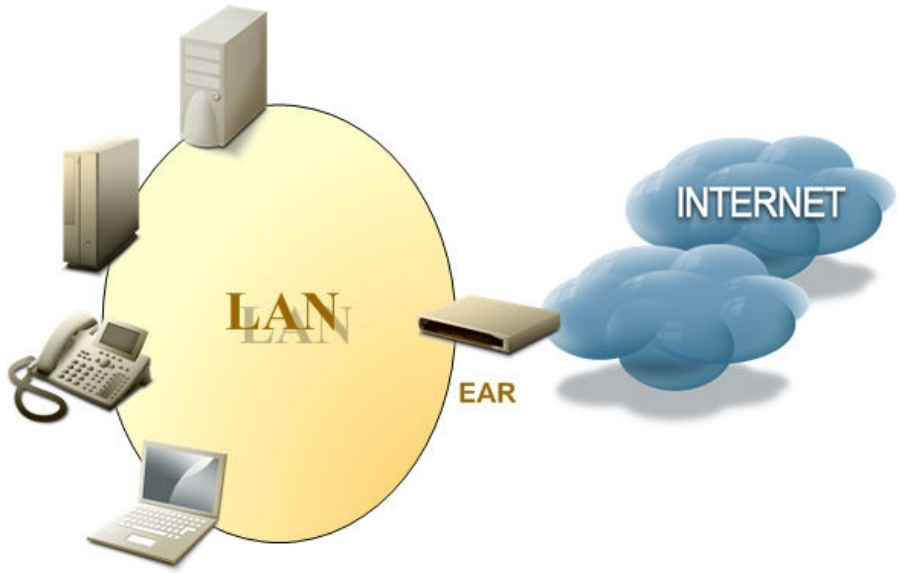

Fig. 19: **Bridge Spoofing**

In this scenario, various devices are connected on the LAN access Internet through the EAR access router. A case arises where a second operator wants to provide backup capability in the event that the EAR router fails, but has to do this without modifying said router configuration or the configuration of the other devices on the LAN. The new operator then inserts a BR backup router between the LAN and the EAR router, as shown in the next figure:

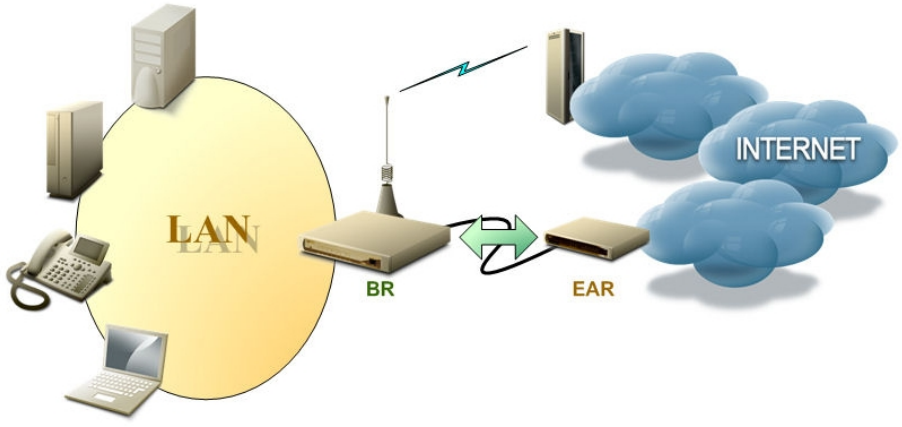

Fig. 20: **Bridge Spoofing**

This new BR router connects to the LAN through an Ethernet port and to the EAR router through another Ethernet port, the EAR communicates with the LAN establishing a Transparent Bridge (STB), as explained in *Using Transparent Bridging (STB)* on page 8. Thus, the new BR backup router can be inserted without modifying the configuration of any of the other devices. However, to provide backup, the BR router must be configured so it processes all the outgoing traffic and sends it through the EAR, or over an alternative link, (the example figure shows a UMTS link) depending on the network conditions. This enables IRB and the Bridge Spoofing feature in the BVI (sub)interface. Consequently, all outgoing traffic is redirected to the BR router and this transmits it over the most convenient path complying with the configured routing criteria.

Where the service needs to be guaranteed, even if the BR router is down, a bypass router, such as the XXX 50 By-

pass, can be used. This has a security mechanism that physically joins the two Ethernet ports (the LAN and EAR ports) when faced with a power failure for example.

The command enabling the Bridge Spoofing feature in the BVI (sub)interface is **spoof ip-address <ear router ip address>**. Through this command, all traffic destined to be routed by the EAR is redirected to the device itself (BVI (sub)interface) instead of being bridged.

Chapter 9 Bridging Configuration

9.1 Accessing the bridging configuration

The main bridge configuration menu is the ASRT menu.

From the ASRT , the main *Virtual Bridge* instance can be configured (identifier is 0). Here new virtual bridge instances can be created, as well as modifying the configuration. When you access a different virtual bridge instance from the main one, the prompt is VBDG.

The main bridge instance (*VIRTUAL BRIDGE 0*) is always created. If you wish to manage an additional instance, you need to create it.

Basically, the configuration options are the same as for a main bridge instance and the rest of the bridging instances with the exception of options relative to BAN and DLS, which are only operative in the main instance.

9.1.1 Accessing the main bridge instance configuration menu (VIRTUAL BRIDGE 0)

To access the main bridge instance configuration menu, run **protocol asrt** from the main configuration menu.

```
Config>protocol asrt

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

To access the NetBIOS configuration commands, run **netbios** from the bridge configuration menu.

```
ASRT config>netbios
-- NetBIOS Support User Configuration --
NetBIOS config>
```

9.1.2 Accessing the bridge virtual instance configuration menu (BRIDGE VIRTUAL)

A virtual bridge consists of an entity that is independent from the bridge, and that device interfaces can be associated with. Each bridge is totally independent of the others; one device interface cannot be shared by different bridge instances. The virtual bridge feature allows you to divide the device into various independent bridges, although physically you only have one device.

To access the configuration menu for a bridge virtual instance, enter **virtual-bridge** followed by the bridge virtual identifier from the main instance configuration menu (ASRT menu).

Example:

```
ASRT config>virtual-bridge 2
-- Virtual ASRT Bridge user configuration --
VBDG config>
```

Note

The configuration options described for the main instance are the same for the virtual instances, with the exception of BAN and DLS options, which are only operative in the main instance. For this reason, in this manual, all the examples given use the main instance configuration menu. Options incompatible with bridge virtual instances are indicated throughout the manual.

Bear in mind that when configuring the main instance, the following prompt appears:

```
ASRT config>
```

Note

And when configuring a virtual instance, the following prompt appears:

```
VBDG config>
```

Note

In the submenus depending on a virtual instance, the VBDG tag must be prefixed.

9.2 Bridging configuration commands

This section describes the bridge configuration commands.

9.2.1 ? (HELP)

Displays the commands available from the current menu. After a specific command, the available options are given.

Syntax:

```
ASRT config>?
```

Example:

```
ASRT config>?
  address                   Add unique station address entries
  ban                       Access to the BAN configuration menu
  bridge                    Enable bridging functionality
  bridge-number             Set bridge number for source routing
  bridge-protocol           Enable protocol for bridging
  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)
  fast-irb                  Enable integrated fast routing and bridging feature
  ibm8209-spanning-tree     Participate in spanning tree protocols with IBM
                            8209
  irb                       Enable integrated routing and bridging feature
  list                      List configuration
  mapping                   Functional address to group address mapping
  name-caching              Access to the Name Caching configuration menu
  netbios                   Access to the Netbios configuration menu
  no                        Negate a command or set its defaults
  port                      Add a LAN/WAN port to the bridging configuration
  protocol-filter           Filter packets based on their protocol type
  route-protocol            Enable protocol for routing
  set                       Configure 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
  ub-encapsulation          Ungermann-Bass OUI encapsulation for XNS frames
  virtual-bridge            Create/enter a Virtual Bridge entity configuration
                            menu
  virtual-segment           Set bridge virtual segment number
  vlan                      Enter 802.1Q bridge menu
  exit
ASRT config>
```

9.2.2 ADDRESS

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 aging settings. Dynamic entries cannot replace permanent entries.

The MAC address for the desired entry must be specified. It can be an individual, multicast, or broadcast address. You can also specify the output forwarding port map for each input port.

Syntax:

```
ASRT config>address <mac-address>
default              Create a new address
source-address-filt  Source Address Filtering Applies
no
       source-address-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
              <output-port>
       different-mapping      Output port mapping for one input port
              <input-port> <output-port>
```

9.2.2.1 DEFAULT

Creates a new permanent entry in the filtering database. This then filters any frames containing this address as a destination address, no matter which port it came from.

Syntax:

```
ASRT config>address <mac-address> default
```

Example:

Creating a new permanent entry to filter all packets with destination MAC address 00-A0-26-00-AC-58.

```
ASRT config>address 00A02600AC58 default
ASRT config>
```

9.2.2.2 SOURCE-ADDRESS-FILT

Allows port-specific address filtering. 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.

Syntax:

```
ASRT config>address <mac-address> source-address-filt
```

Example:

Creating a new permanent entry to filter all packets generated by the station with MAC address 00-A0-26-00-AC-58.

```
ASRT config>address 00A02600AC59 source-address-filt
ASRT config>
```

9.2.2.3 NO

Negates a command or sets the default value option.

9.2.2.3.1 SOURCE-ADDRESS-FILT

Disables source address traffic filtering.

Syntax:

```
ASRT config>address <mac-address> no source-address-filt
```

Example:

Allows packet bridging for previously filtered packets from address 00-A0-26-AC-5.

```
ASRT config>address 00A02600AC59 no source-address-filt
ASRT config>
```

9.2.2.4 BRIDGE

Specifies which ports filtering is performed on for a permanent filtering entry. It's possible to define ports that allow bridging of frames with a particular destination address. To do this, define port mapping indicating for each input port which output ports can perform frame bridging. The different available suboptions are listed further on in this section.

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).

It's only possible to introduce a bridge option by address. If you want to modify the configuration selected for an address, first eliminate the filtering entry and then re-create it with the required option.

Syntax:

```
ASRT config>address <mac-address> bridge ?
  all-same-port       Use all output port mapping for all input Ports
  same-mapping        Use same output port mapping for all input Ports
  different-mapping    Output port mapping for one input port
```

9.2.2.4.1 ALL-SAME-PORT

Creates *one* output port map for a MAC address, for all input ports rather than only mapping to specific ports.

Syntax:

```
ASRT config> address <mac-address> bridge all-same-port
```

Example:

```
ASRT config>address 000000334455 bridge all-same-port
ASRT config>
```

9.2.2.4.2 SAME-MAPPING

Creates port mapping for a MAC address that includes all input ports for an output port. When a frame containing said address is received (regardless of the input port), it's forwarded to all output forwarding ports specified through this option, with the exception of the input port.

For the same address, enter this command as many times as you consider necessary to map the output ports.

Syntax:

```
ASRT config>address <mac-address> bridge same-mapping <out-port>
```

out-port	Bridge output port. This can take values between 1 and 254.

Example:

Creates a filter entry for MAC address 00-00-00-33-44-5 so traffic destined for this address is sent through ports 1 and 2, independently of the input port.

```
ASRT config>address 000000334455 bridge same-mapping 1
ASRT config>address 000000334455 bridge same-mapping 2
ASRT config>
```

9.2.2.4.3 DIFFERENT-MAPPING

This is the most generic option to set up port mapping with a MAC address. This allows a MAC address to indicate which ports can be used as output ports for each input port.

For the same address, enter this command as many times as necessary to map the required ports.

Syntax:

```
ASRT config>address <mac-address> bridge different-mapping <in-port> <out-port>
```

In-port	Bridge input port. This can take values between 1 and 254.
out-port	Bridge output port. This can take values between 1 and 254.

Example:

Creates a filter entry for MAC address 00-00-00-33-44-55, so traffic destined for this address is sent through ports 1 and 2 when it enters through port 3, and through port 3 if it enters through port 1 or 2.

```
ASRT config>address 000000334455 bridge different-mapping 3 1
ASRT config>address 000000334455 bridge different-mapping 3 2
ASRT config>address 000000334455 bridge different-mapping 1 3
ASRT config>address 000000334455 bridge different-mapping 2 3
ASRT config>
```

The following sections present examples of how to use the **address** command to manage address table entries.

Example 1:

Enabling destination address filtering for an entry.

```
ASRT config>address 000000334455 default
ASRT config>
```

After adding the address, verify the status by running **list range**.

Syntax:

```
ASRT config>list range < Start-Index> < Stop-index >
```

The value for the beginning and end of the index is included in the interval [1..65535].

The example below shows no port map exists for said entry (in bold) and Destination Address Filtering (DAF) is enabled.

```
ASRT config>list range 1 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>
```

Example 2:

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

```
ASRT config>address 000000012345 bridge different-mapping 1 1
ASRT config>address 000000012345 bridge different-mapping 1 2
ASRT config>address 000000012345 bridge different-mapping 2 1
ASRT config>address 000000012345 bridge different-mapping 2 2
ASRT config>address 000000012345 source-address-filt
ASRT config>
```

After adding the address, verify the status by running **list range**. The example below shows an entry (in **bold**) with ports 1 and 2 as input ports and with separate port maps for both input ports. Source Address Filtering (SAF) is also enabled.

```
ASRT config>list range 1 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>
```

Example 3:

Creating a single output port map for all input 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 running **list range**. The example below shows an entry (in **bold**) that has a single port map for all input ports. Source Address Filtering (SAF) is also enabled.

```
ASRT config>list range 1 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>
```

9.2.3 BAN

This accesses the BAN parameter configuration menu. For further information on how to configure BAN, please see manual bintec *Dm716-I DLSw Protocol* .

This menu is only accessible through the ASRT menu, i.e., from the menu associated with the main bridge entity. BAN is not configurable in VBDG menus associated with other virtual bridge entities.

Syntax:

```
ASRT config>ban
```

Example:

```
ASRT config>ban

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

9.2.4 BRIDGE

Enables transparent bridging.

Example:

```
ASRT config>bridge
ASRT config>
```

Command history:

Release	Modification
11.01.00	This command is obsolete in version 11.01.00. To enable transparent bridging, run **add device bvibvi#**.

9.2.5 BRIDGE-NUMBER

Changes the bridge number used by the bridge in source routing.

Syntax:

```
ASRT config>bridge-number <bridge-number>
```

Example:

```
ASRT config>bridge-number ?
  <hex 1..f>    Bridge number
ASRT config>bridge-number A
```

9.2.6 BRIDGE-PROTOCOL

Enables bridge for a protocol. By default bridge is enabled, this command is used to eliminate the configuration achieved through **no bridge-protocol**, used to disable the bridge for a protocol.

Syntax:

```
ASRT config>bridge-protocol ?
  ip    IP protocol group
  ipv6    IPv6 protocol group
ASRT config>
```

Example:

Run **bridge-protocol ip** to enable the bridge for packets from the IPv4 protocol group.

```
ASRT config>bridge-protocol ip
ASRT config>
```

Run **bridge-protocol ipv6** to enable the bridge for packets from the IPv6 protocol group.

```
ASRT config>bridge-protocol ipv6
```

```
ASRT config>
```

9.2.7 DLS

Enables DLSw over the bridge. The router running DLSw appears to be a bridge to the end stations.

This command is only accessible through the ASRT menu, i.e., from the menu associated with the main bridge entity. DLSw is not configurable in VBDG menus associated with other virtual bridge entities.

Example:

```
ASRT config>dls
ASRT config>
```

9.2.8 DUPLICATE

Enables the generation of duplicate STE (Spanning Tree Explorer) or TSF (Transparent Spanning Frames) frames. Duplicate frame generation is enabled by default; this command is used to eliminate the configuration obtained by running **no duplicate**. The **duplicate** command must be followed by the frame type identifier (TSF or STE) and the port this affects.

Activates 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 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).

Run **duplicate ste** to tell the bridge to send spanning tree explorer frames created for the source-routing environment. Run **duplicate tsf** to tell the bridge to send 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>duplicate <type> <port>
```

9.2.8.1 DUPLICATE STE

Example:

```
ASRT config>duplicate ste 2
ASRT config>
```

9.2.8.2 DUPLICATE TSF

Example:

```
ASRT config>duplicate tsf 1
ASRT config>
```

9.2.9 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* on page 21 in *Miscellaneous Bridge Features* on page 21 for more details.

Syntax:

```
ASRT config>ethertype-ibmrt-pc <Port Number>
```

Example:

```
ASRT config>ethertype-ibmrt-pc 1
ASRT config>
```

9.2.10 FA-GA-MAPPING

Enables assigning of group addresses to functional addresses and vice versa. This feature is needed 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 addresses is inevitable. Mapping is enabled by default if mapping addresses have been added.

Example:

```
ASRT config>fa-ga-mapping
ASRT config>
```

9.2.11 FAST-IRB

Enables fast integrated routing and bridge (IRB). IRB differs from basic features where some functional bridge blocks deactivate when the network topology analysis (STP) determines only one of the bridge ports is in a Forwarding state. This produces better device performance regarding switch capacity.

For further information on IRB, please see *Integrated Routing and Bridging* on page 27 in this manual.

Syntax:

```
ASRT config>fast-irb
```

Command history:

Release	Modification
11.01.00	This command is obsolete from version 11.01.00. The **route-protocol** command automatically enables fast IRB.

9.2.12 IBM8209_SPANNING_TREE

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

Example:

```
ASRT config>ibm8209-spanning-tree
ASRT config>
```

9.2.13 IRB

Enables the *integrated routing and bridging* feature. For further information, please see *Integrated Routing and Bridging* on page 27 in this manual.

Syntax:

```
ASRT config>irb
```

Command history:

Release	Modification
11.01.00	This command is obsolete from version 11.01.00. The **route-protocol** command automatically enables fast IRB, which is compatible with integrated routing and bridging.

9.2.14 LIST

Displays information on the complete bridge configuration, or on selected configuration parameters.

Syntax:

```
ASRT config>list ?
  address           Reads an address entry from the permanent database
  bridge            Lists all general information regarding the bridge
  filtering         Displays the parameters associated to the bridge filter
  mapping           Lists specific address mapping for given protocol
  permanent         Displays the number of entries in the bridge's database
  port              Displays port information related to ports already configured
```

```
  prot-filter       Reads a current list of the filter protocol types
  range             Reads a range of address entries from the permanent database
  spanning-tree     Bridge information related to the spanning tree protocol
  virtual-bridge    Virtual Bridge entities
```

9.2.14.1 LIST ADDRESS

Reads an address entry from the permanent database.

Syntax:

```
ASRT config>list address <mac-address>
```

Example:

```
ASRT config>list address 000000123456
ADDRESS                 ENTRY TYPE      PORT MAP
=======                 ==========      ========
00-00-00-12-34-56       PERMANENT       Input Port:  ALL PORTS
                                        Output ports:  1, 2
ASRT config>
```

Example:

```
ASRT config>list address 000000123456
ADDRESS                 ENTRY TYPE      PORT MAP
=======                 ==========      ========
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 standard for future use. Frames to reserved addresses are discarded.
	Registered	The entry is for the bridge itself.
	SAF	Appears after the entry type if source address filtering is configured.
Input Port	The numbers of input port(s) associated with that address entry.	
Output Port	The numbers of output port(s) associated with said address entry. NONE/DAF indicates destination address filtering applies because no ports have been selected to be associated with said address entry.	

9.2.14.2 LIST BRIDGE

Lists all general information on the bridge.

Example:

```
ASRT config>list bridge

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


Virtual Bridge ID: 0
Bridge:    Enabled                              Bridge behavior: STB
                   +---------------------------------------+
-------------------|       SOURCE ROUTING INFORMATION      |-----------------
                   +---------------------------------------+
```

```
Bridge Number:          00                      Segments:          0
Max ARE Hop Cnt:        00                      Max STE Hop cnt:  00
1:N SRB:                Not Active              Internal Segment:  0x000
LF-bit interpret:       Extended
                   +---------------------------------------+
-------------------|             SR-TB INFORMATION          |-----------------
                   +---------------------------------------+
SR-TB Conversion:        Disabled
TB-Virtual Segment:      0x000                  MTU of TB-Domain:  0
                   +---------------------------------------+
-------------------|    SPANNING TREE PROTOCOL INFORMATION   |------------------
                   +---------------------------------------+
Bridge Address:          Default          STP Participation:       IEEE802.1D
Bridge Priority:         32768/0x8000 (VLAN 1)
Bridge Priority:         32768/0x8000 (VLAN 55)
Bridge Priority:         32768/0x8000 (VLAN 88)
                   +---------------------------------------+
-------------------|        TRANSLATION INFORMATION          |------------------
                   +---------------------------------------+
FA<=>GA Conversion:      Enabled                UB-Encapsulation:  Disabled
DLS for the bridge:      Disabled
                   +---------------------------------------+
-------------------|            PORT INFORMATION            |-------------------
                   +---------------------------------------+
Number of ports added: 2
Port:   1    Interface:     ethernet0/0 Behavior:    STB Only   STP: Enabled   (VLAN 1)
                                                                    Enabled   (VLAN 55)
                                                                    Enabled   (VLAN 88)

Port:   2    Interface:     ethernet0/1 Behavior:    STB Only   STP: Enabled   (VLAN 1)
                                                                    Enabled   (VLAN 88)
                                                                    Enabled   (VLAN 55)



ASRT config>
```

Virtual Bridge ID	Virtual bridge identifier.
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	Bridge priority, used in the Spanning Tree protocol. Furthermore, if the PVST is enabled, it will appear a line per VLAN configured.
Bridge Number	Bridge number. Used in source routing bridge. 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 forwarding 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 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 the bridge participates in.
Number of ports added	The number of ports configured on the bridge.
Port Number	Bridge port identifier.

Interface	Interface used for bridging. Add at least two interfaces to participate in bridging.
Port Behavior	Indicates the method of bridging being used by the port. The values are STB for Transparent, SRB for Source Routing and SR-TB for Source Routing-Transparent conversion bridging.
Port STP	Indicates if the STP is enabled or not for each VLAN configured in the port.

9.2.14.3 LIST FILTERING

Displays the parameters associated with 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 the bridge filtering database can have.
Ageing Time:	Time after which the dynamic entries in the filtering database disappear.
Resolution:	Temporary resolution used for the expiry of the dynamic entries in the filtered database.

9.2.14.4 LIST MAPPING

Lists specific address mapping for a given protocol.

Syntax:

```
ASRT config>list mapping ?
  dsap     Specific functional address to group address mapping for a DSAP id
  ether    Specific functional address to group address mapping for an ether id
  snap     Specific functional address to group address mapping for a SNAP id
```

9.2.14.4.1 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>
```

9.2.14.4.2 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>
```

9.2.14.4.3 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>
```

9.2.14.5 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>
```

9.2.14.6 LIST PORT

Displays port information on preconfigured ports. If a port number is not specified, information on all ports is displayed.

Example:

```
ASRT config>list port
Port Id (dec)    : 128: 1, (hex): 80-01     (VLAN 1)
Port Id (dec)    : 128: 1, (hex): 80-01     (VLAN 55)
Port Id (dec)    : 128: 1, (hex): 80-01     (VLAN 88)
Port State       : Enabled
STP Participation: Enabled
Port Supports    : Transparent Bridging Only
Assoc Interface  :     ethernet0/0
Path Cost        : 0     (VLAN 1)
Path Cost        : 0     (VLAN 55)
Path Cost        : 0     (VLAN 88)
------------------------------------------------------------------------------
Port Id (dec)    : 128: 2, (hex): 80-02     (VLAN 1)
Port Id (dec)    : 128: 2, (hex): 80-02     (VLAN 88)
Port Id (dec)    : 128: 2, (hex): 80-02     (VLAN 55)
Port State       : Enabled
STP Participation: Enabled
Port Supports    : Transparent Bridging Only
Assoc Interface  :     ethernet0/1
Path Cost        : 0     (VLAN 1)
Path Cost        : 0     (VLAN 88)
Path Cost        : 0     (VLAN 55)
------------------------------------------------------------------------------
ASRT config>
```

Port ID	Port identifier. The ID consists of two parts: port priority and port number. In the example, 128 is the priority and 1 or 2 is the port number. In hexadecimal format, the low-order byte denotes the port number and the high order byte denotes priority. At the end, the VLAN configured in the port appears. Furthermore, if the PVST is enabled, it will appear a line per VLAN configured in a port.
Port State	Shows if the port is enabled or disabled.
STP Supports	Shows whether or not the port participates in the Spanning Tree protocol.
Port Supports	Displays the 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 Spanning Tree Explorer Frames transmission 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. For FR circuits, this also indicates the circuit name.
Path Cost	Cost associated with the port used for the Spanning Tree protocol for possible root path cost. The range is 1 to 65535. Furthermore, if the PVST is enabled, it will appear a line per VLAN configured in a port.

Note

If IBM RT-PC Ethertype processing is enabled, they appear on this display (otherwise they don't).

9.2.14.7 LIST PROT-FILTER

Displays the configured protocol filters. If you do not specify a port number, information on all ports is displayed.

Example:

```
ASRT config>list prot-filter
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>
```

Protocol Class	Displays protocol class (SNAP, Ethernet, or DSAP).
Protocol Type	Protocol ID in hexadecimal format.
Protocol State	Indicates the protocol is being filtered.
Port Map	Ports where protocol filtering is applied. This field appears when you execute **list prot-filter** without specifying a port.

9.2.14.8 LIST RANGE

Displays a range of address entries from the permanent database. Run **list permanent** to determine the number of entries in the database.

Syntax:

```
ASRT config>LIST RANGE  <start-index>  <stop-index>
```

Example:

```
ASRT config>list range 17 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>
```

The following explains the various fields:

Address	6-byte MAC address the entry is associated with.	
Entry Type	Specifies one of the following types:	
	Reserved	Address reserved by the IEEE802.1D standard.
	Registered	Addresses internally registered for the bridge so it works correctly.
	Permanent	Entries permanently created in the configuration process. These entries are not deleted when the device is switched off and on.
	Perm/SAF	Permanent entries with source address filtering.
	Free	Free entries in the database, not associated with any MAC address.
Port Map	Port map associated with the entry. This indicates the output port for each input port through which a destination address associated with the entry can be sent. Where a port map is not defined, NONE/DAF is displayed indicating filtering is executed by the destination address.	

9.2.14.9 LIST SPANNING-TREE

Displays information of each instance of the spanning tree protocol.

Example of only one instance:

```
ASRT config>list spanning-tree
Bridge Identifier               : 32768/000000000000     (VLAN 1) (using port address)
```

```
Bridge-Max-Age (in seconds)      : 20
Bridge-Hello-Time (in seconds)   : 2
Bridge-Forward-Delay (in seconds): 15
TxHoldCount (in seconds)         : 6
Protocol Version                 : RSTP normal operation
Bridge Identifier                : 32768/000000000000     (VLAN 55) (using port address)
Bridge-Max-Age (in seconds)      : 20
Bridge-Hello-Time (in seconds)   : 2
Bridge-Forward-Delay (in seconds): 15
TxHoldCount (in seconds)         : 6
Protocol Version                 : RSTP normal operation
Bridge Identifier                : 32768/000000000000     (VLAN 88) (using port address)
Bridge-Max-Age (in seconds)      : 20
Bridge-Hello-Time (in seconds)   : 2
Bridge-Forward-Delay (in seconds): 15
TxHoldCount (in seconds)         : 6
Protocol Version                 : RSTP normal operation
ASRT config>
```

Bridge Identifier	Bridge Identifier. The bridge identifier is made up of two fields: one 2-byte field indicating priority and one 6-byte field, the bridge MAC address. When the bridge address consists of six zeros, the device (on booting) selects the MAC address from one of its ports and uses this as the bridge address. The bridge identifier is used to select the root bridge in the Spanning Tree protocol. At the end, the VLAN configured in the port appears.
Bridge-Max-Age	Maximum age (period of time) that should be used to time out spanning-tree-protocol-related information.
Bridge-Hello-Time	Time between Hello BPDUs.
Bridge-Forward-Delay	Time period used before changing to another state in a port (should this bridge become the root).
TxHoldCount	Maximum number of BPDUs that can be sent through a port in one second.
Protocol Version	Spanning Tree protocol version currently running in the bridge. This can be *STP compatibility* if it forces a version of Spanning Tree compatible with an old version of Spanning Tree protocol, or *RSTP normal operation* if it is running Rapid Spanning Tree protocol.

9.2.14.10 LIST VIRTUAL-BRIDGE

Lists the virtual bridge instances in the device, as well as the ports associated with each of them.

Instance 0, corresponding to the main instance, is always displayed.

This command is only available in the main bridge instance configuration menu (ASRT menu).

Example:

```
ASRT config>list virtual-bridge
Virt. Bridge ID                        Associated Interfaces
---------------  ------------------------------------------------------------
      0          ethernet0/0 ethernet0/1
      1          wlan2/0 serial0/0
ASRT config>
```

9.2.15 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 needed for bidirectional mapping.

Syntax:

```
ASRT config>mapping <dlh-type> <Protocol-Type> <Group-Address> <Functional-Address>
```

dlh-type	Data Link Header Type. The available options are **dsap** (Destination Service Address Point), **ether** (Ethertype) or **snap** (Subnetwork Access Protocol).
Protocol-type	Protocol type. Where the **dsap** is configured, the DSAP protocol type is a value in the hexadecimal range from 1 to FE. Where **ether** is configured, the Ethernet pro-

	tocol type is a value in the hexadecimal range from 5DD to FFFF. When **snap** is configured, the SNAP protocol type is a 10 hexadecimal digit value.
group-address	6-byte (12-digit hexadecimal) group/multicast address.
functional-address	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 as follows:

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 1 ab0000020000 c00020000000
ASRT config>
```

Example 2:

```
ASRT config>mapping ether 6002 ab0000020000 c00020000000
ASRT config>
```

Example 3:

```
ASRT config>mapping snap 0000006003 ab0000030000 c00010000000
ASRT config>
```

9.2.16 NAME-CACHING

Accesses the Name Caching configuration menu and the duplicate frame filtering for NetBIOS.

Syntax:

```
SRT config>name-caching

Name Cache Config>
```

The name cache feature considerably reduces the number of bridged Name-Query frames.

NetBIOS uses 16 character names to identify devices. The first step in data transfer is for the client to obtain a physical address from the server name. To do this, the client sends a Spanning Tree explorer frame known as Name-Query. The server responds with a Name-Query-Response, which contains its MAC address and the route to reach it.

With the name cache, the bridge maintains a database of names and routes. Each time a Name-Query-Response frame is received, the MAC address and route are extracted and stored in the database.

Consequently, when the bridge receives a Name-Query frame, it checks if the queried name is in its database. If it is, it converts the STE frame to an SRF frame. The entries in the database timeout comply with a configurable timer.

The process carried out by the cache on receiving a Name-Query frame is as follows:

(1) It searches the database for the name being queried.

(2) If the name is not found in the database, the frame is sent as is.

(3) If an entry associated with the name, indicating that a response has been received, is found, the time interval is updated and the frame is sent converting it into an SRF using the information stored in the entry.

(4) If a Name-Query-Response has not been received from the server within the required time, the entry is invalidated and the frame is sent as is (i.e., as an STE frame).

The process carried out by the cache on receiving a Name-Query-Response frame is as follows:

(1) If there is an entry in the database for this name, the received information and the time of the last response is updated indicating a response has been received.

(2) If there isn't an entry in the database for this name, it is created with the received information.

Moreover, the names cache permits duplicate frame filtering. The Name-Query, Add-Name and Add-Group-Name frames are sent up to six times. The duplicated frame filtering feature can be specified so instance bridging is only executed for each type of frame in the time specified by the user.

The following are the commands available in the NetBIOS name cache configuration.

Command	Function
? (HELP)	Displays all the configuration commands, or lists options for specific commands.
DISABLE	Disables Name-caching feature or duplicate frame filtering.
ENABLE	Enables Name-caching feature or duplicate frame filtering.
LIST	Displays the currently implemented Name-caching configurations.
PORT	Selects the port for configuration purposes.
TIMER	Sets the different timers used in the name cache and the duplicated frame filtering.
EXIT	Exits the Name-caching and duplicated frame filtering configuration menu.

9.2.16.1 ? (HELP)

Use the **?** (HELP) command to list the available commands. Enter this command after a specific command to see the options.

Example:

```
Name Cache Config>?
  disable    Disable name-caching facility and duplicate frame filtering
  enable     Enable name-caching facility and duplicate frame filtering
  list       List configuration
  port       Selects a specific interface for configuring purposes
  timer      Configure protocol timers
  exit
Name Cache Config>
```

9.2.16.2 DISABLE

Disables Name-caching feature or duplicate frame filtering.

Syntax:

```
Name Cache Config>disable ?
  add-name-filtering    Disable duplicate frame filtering
  name-caching          Disable name-caching facility
Name Cache Config>
```

9.2.16.2.1 DISABLE ADD-NAME-FILTERING

Disables duplicate frame filtering.

Example:

```
Name Cache Config>disable add-name-filtering
Name Cache Config>
```

9.2.16.2.2 DISABLE NAME-CACHING

Disables Name-caching feature.

Example:

```
Name Cache Config>disable name-caching
Name Cache Config>
```

9.2.16.3 ENABLE

Enables Name-caching feature or duplicate frame filtering.

Syntax:

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

9.2.16.3.1 ENABLE ADD-NAME-FILTERING

Enables duplicate frame filtering. A timer is used to ensure that bridging is only carried out on an instance for each of the Name-Query, Add-Name and Add-Group-Name frames in the specified time period.

Example:

```
Name Cache Config>enable add-name-filtering
Name Cache Config>
```

9.2.16.3.2 ENABLE NAME-CACHING

Enables Name-caching feature.

Example:

```
Name Cache Config>enable name-caching
Name Cache Config>
```

9.2.16.4 LIST

Displays the current configuration associated with the name cache and the duplicated frame filtering.

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>
```

9.2.16.5 PORT

Accesses the name cache and the duplicated frame filtering for a certain port configuration submenu.

Example:

```
Name Cache Config>port 2
Name Cache Port Config>
```

The following commands are available in the port submenu:

Syntax:

```
Name Cache Port Config>?
  disable    Disable name-caching facility and duplicate frame filtering
  enable     Enable name-caching facility and duplicate frame filtering
  list       List configuration
  exit
```

The meaning of these commands and their options is the same as in the global menu, except the parameters refer to a particular port instead of being global parameters.

If, for example, you want to disable duplicated frame filtering in port 3, execute the following commands from the global configuration menu.

```
Name Cache Config>port 3

Name Cache Port Config>
Name Cache Port Config>disable add-name-filtering
Name Cache Port Config>exit
Name Cache Config>
```

9.2.16.6 TIMER

Configures the different timers used in the name cache and duplicated name filtering.

Syntax:

```
Name Cache Config> timer ?
  add-name           Set the time within which duplicate frames are filtered
  entry              Set the entry idle timer
  server-response    Set the server timer
Name Cache Config>timer
```

9.2.16.6.1 TIMER ADD-NAME

Sets the time within which duplicate frames are filtered. Default is 7 seconds.

Syntax:

```
Name Cache Config>timer add-name <1s..32s>
```

Example:

```
Name Cache Config>timer add-name 27s
Name Cache Config>
```

9.2.16.6.2 TIMER ENTRY

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

Syntax:

```
Name Cache Config>timer entry <10s..18h12m15s>
```

Example:

```
Name Cache Config>timer entry 455s
Name Cache Config>
```

9.2.16.6.3 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 becomes invalid. Default is 3 seconds.

Syntax:

```
Name Cache Config>timer server-response <1s..16s>
```

Example:

```
Name Cache Config>timer server-response 10
Name Cache Config>
```

9.2.16.7 EXIT

Exits the name cache and frame duplication configuration menu and returns to the bridge configuration menu.

Example:

```
Name Cache Config>exit
ASRT config>
```

9.2.17 NETBIOS

Accesses the NetBIOS configuration menu.

See *NetBIOS Filtering and Caching commands* on page 105, for an explanation for the NetBIOS configuration commands.

Syntax:

```
ASRT config>netbios
```

Note

If you have not purchased the NetBIOS feature, the following message appears if you use the command:

NetBIOS Support not in load.

9.2.18 NO

Configures the parameters with their default values or deletes the configuration.

Syntax:

```
ASRT config>no ?
  address                   Add unique station address entries
  bridge                    Enable bridging functionality
  bridge-protocol           Enable protocol for bridging
  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)
  fast-irb                  Enable integrated fast routing and bridging feature
  ibm8209-spanning-tree     Participate in spanning tree protocols with IBM
                            8209
  irb                       Enable integrated routing and bridging feature
  mapping                   Functional address to group address mapping
  port                      Add a LAN/WAN port to the bridging configuration
  protocol-filter           Filter packets based on their protocol type
  route-protocol            Enable protocol for routing
  set                       Configure 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
  ub-encapsulation          Ungermann-Bass OUI encapsulation for XNS frames
  virtual-bridge            Create/enter a Virtual Bridge entity configuration
                            menu
ASRT config>
```

9.2.18.1 NO ADDRESS

Deletes a MAC address entry form from the permanent database.

Syntax:

```
ASRT config>no address <mac-address>
```

You cannot delete reserved multicast addresses. If you attempt to delete an address entry that does not exist, the following error message is displayed:

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

Example:

```
ASRT config>no address 001122334455
ASRT config>
```

9.2.18.2 NO BRIDGE

Completely disables the bridge functionality. This does not eliminate the value of the previously configured parameters.

Example:

```
ASRT config>no bridge
ASRT config>
```

Command history:

Release	Modification
11.01.00	This command is obsolete from version 11.01.00. Use the command **no devicebyvi#** to disable the bridge.

9.2.18.3 NO BRIDGE-PROTOCOL

Disables bridge for a protocol so it doesn't bridge received packets pertaining to the specified protocol.

Syntax:

```
ASRT config>no bridge-protocol ?
  ip    IP protocol group
  ipv6    IPv6 protocol group
ASRT config>
```

Example:

Run **no bridge-protocol ip** to disable bridging of packets from the IPv4 protocol group.

```
ASRT config>no bridge-protocol ip
ASRT config>
```

Run **no bridge-protocol ipv6** to disable bridging of packets from the IPv6 protocol group.

```
ASRT config>no bridge-protocol ipv6
ASRT config>
```

9.2.18.4 NO DLS

Disables DLSw over bridge.

Example:

```
ASRT config>no dls
ASRT config>
```

9.2.18.5 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 (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 disabling the creation of one of these types of frames. Said command does not allow you to disable both types of frames simultaneously.

Running **no duplicate ste** tells the bridge to refrain from sending spanning tree explorer frames created for the source-routing environment. Running **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.

Deactivates the creation of duplicate frames in mixed bridge environments. The SR-TB in an 802.5 interface (with active source routing and transparent bridging) can create inconsistencies when frame bridging is executed for an unknown source or multicast. The bridge does not know if the destination is behind source routing bridge (only) or a transparent bridge.

The **no duplicate** command is applied by the bridge port.

Syntax:

```
ASRT config>no duplicate <type> <port>
```

9.2.18.5.1 NO DUPLICATE STE

Example:

```
ASRT config>no duplicate ste 1
ASRT config>
```

9.2.18.5.2 NO DUPLICATE TSF

Example:

```
ASRT config>no duplicate tsf 2
ASRT config>
```

9.2.18.6 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 *Miscellaneous Bridge Features* on page 21, *IBM RT feature for SNA traffic* on page 21.

The **no ethertype-ibmrt-pc** command is applied by the bridge port.

Syntax:

```
ASRT config>no ethertype-ibmrt-pc <port>
```

Example:

```
ASRT config> no ethertype-ibmrt-pc 1
ASRT config>
```

9.2.18.7 NO FA-GA-MAPPING

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

Example:

```
ASRT config>no fa-ga-mapping
ASRT config>
```

9.2.18.8 NO FAST IRB

Disables fast integrated routing and bridging features (IRB). For further information on IRB, please see *Integrated Routing and Bridging* on page 27 in this manual.

Command history:

Release	Modification
11.01.00	This command is obsolete from version 11.01.00. The fast integrated routing and bridging feature is disabled when all the routing protocols have been removed through the **no route-protocol** command.

9.2.18.9 NO IBM8209_SPANNING_TREE

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

Example:

```
ASRT config>no ibm8209-spanning-tree
ASRT config>
```

9.2.18.10 NO IRB

Disables integrated routing and bridging. For further information, please see *Integrated Routing and Bridging* on page 27 in this manual.

Syntax:

```
ASRT config>no irb
```

Command history:

Release	Modification
11.01.00	This command is obsolete from version 11.01.00. The integrated routing and bridging feature is disabled when all routing protocols have been removed using the **no route-protocol** command.

9.2.18.11 NO MAPPING

Eliminates assigning a functional address to a group address for a specific protocol identifier.

Syntax:

```
ASRT config>no mapping <dlh-type> <protocol-type> <group-address>
```

dlh-type	Data Link Header Type. The available options are **dsap** (Destination Service Address Point), **ether** (Ethertype) or **snap** (Subnetwork Access Protocol).
Protocol-type	Protocol type. Where the **dsap** is configured, the DSAP protocol type is a value in the hexadecimal range from 1 to FE. Where **ether** is configured, the Ethernet protocol type is a value in the hexadecimal range from 5DD to FFFF. When **snap** is configured, the SNAP protocol type is a 10 hexadecimal digit value.
group-address	6-byte (12-digit hexadecimal) group/multicast address.

Example 1:

```
ASRT config>no mapping dsap fe ab0000020000
ASRT config>
```

Example 2:

```
ASRT config>no mapping ether 0800 ab0000020000
ASRT config>
```

Example 3:

```
ASRT config>no mapping snap 0000006002 ab0000020000
ASRT config>
```

9.2.18.12 NO PORT

Eliminates a port from the bridge configuration.

Syntax:

```
ASRT config>no port <port>
```

Example:

```
ASRT config>no port 1
ASRT config>
```

9.2.18.13 NO PROTOCOL-FILTER

Eliminates a previously created protocol filter.

Syntax:

```
ASRT config>no protocol-filter <dlh-type> <protocol-type> <port>
```

dlh-type	Data Link Header Type. The available options are **dsap** (Destination Service Address Point), **ether** (Ethertype) or **snap** (Subnetwork Access Protocol).
Protocol-type	Protocol type. Where the **dsap** is configured, the DSAP protocol type is a value in the hexadecimal range from 1 to FE. Where **ether** is configured, the Ethernet protocol type is a value in the hexadecimal range from 5DD to FFFF. When **snap** is configured, the SNAP protocol type is a 10 hexadecimal digit value.
Port	Bridge port where the filter is applied.

Example 1:

```
ASRT config>no protocol-filter dsap 1 1
ASRT config>
```

Example 2:

```
ASRT config>no protocol-filter ether FFFF 1
ASRT config>
```

Example 3:

```
ASRT config>no protocol-filter snap 0000000800 1
ASRT config>
```

9.2.18.14 NO SET

Eliminates the configuration created through the **set** command.

Syntax:

```
ASRT config>no set ?
  spanning-tree    Modifies the spanning-tree parameters
```

Example:

Disabling the default configuration for BPDU filtering.

```
ASRT config>no set spanning-tree bpdu-filter default
```

9.2.18.15 NO ROUTE-PROTOCOL

Disables routing for a protocol.

Syntax:

```
ASRT config>no route-protocol ?
  ip    IP protocol group
  ipv6    IPv6 protocol group
```

Example:

```
ASRT config>no route-protocol ip
ASRT config>
```

9.2.18.16 NO SOURCE-ROUTING

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

Syntax:

```
ASRT config>no source-routing <port>
```

Example:

```
ASRT config>no source routing 1
ASRT config>
```

9.2.18.17 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 automatically complies with the transparent spanning tree.

Syntax:

```
ASRT config>no spanning-tree-explorer <port>
```

Example:

```
ASRT config>no spanning-tree-explorer 1
ASRT config>
```

9.2.18.18 NO SR-TB-CONVERSION

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

Syntax:

```
ASRT config>no sr-tb-conversion
```

9.2.18.19 NO STP

Deactivates Spanning Tree protocol participation for the entire bridge.

Syntax:

```
ASRT config>no stp
```

9.2.18.20 NO TRANSPARENT

Disables transparent bridging feature on the given port. This command is useful when an alternative communication method, such as source routing, is desirable.

Also this command is used to enable SRB and SR-TB. This command has pitfalls, so use it with care. For instance, using it on an Ethernet interface disables bridging for that interface.

Syntax:

```
ASRT config>no transparent <port>
```

Example:

```
ASRT config>no transparent 1
ASRT config>
```

9.2.18.21 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.

Syntax:

```
ASRT config>no ub-encapsulation
```

9.2.18.22 NO VIRTUAL-BRIDGE

Eliminates a previously created virtual bridge.

Syntax:

```
ASRT config>no virtual-bridge <entity>
```

Example:

```
ASRT config>no virtual-bridge 1
ASRT config>
```

9.2.19 PORT

Adds a LAN/WAN interface to the bridging configuration. Associates a port number with the interface and enables said port participation in transparent bridging. If you add a Frame Relay interface, you must also specify the circuit name.

Note

You cannot add an interface corresponding to an Ethernet switch where one of its ports has Spanning Tree Protocol enabled through the **port <id> stp enable instance <id>** command (on the switch configuration menu).

Syntax:

```
ASRT config>port <interface-name> <port-number> [<circuit-name>]
```

Example 1:

```
ASRT config>port ethernet0/1 2
ASRT config>
```

Example 2:

```
ASRT config>port fr1 3 Prueba-01
```

```
ASRT config>
```

9.2.20 PROTOCOL-FILTER

Creates protocol filters. The bridge filters packets based on their protocol type. It also discards matching ARP packets.

Syntax:

```
ASRT config>protocol-filter <dlh-type> <protocol-type> <port>
```

dlh-type	Data Link Header Type. The available options are **dsap** (Destination Service Address Point), **ether** (Ethertype) or **snap** (Subnetwork Access Protocol).
Protocol-type	Protocol type. Where the **dsap** is configured, the DSAP protocol type is a value in the hexadecimal range from 1 to FE. Where **ether** is configured, the Ethernet protocol type is a value in the hexadecimal range from 5DD to FFFF. Where **snap** is configured, the SNAP protocol type is a 10 hexadecimal digit value.
Port	Bridge port where the filter is applied.

For Ethernet, it's also possible to configure the configured filters function through this command

Syntax:

```
ASRT config>protocol-filter <inclusive | exclusive>
```

If you configure **protocol-filter inclusive**, bridge is only executed for inclusive packets in the configured filters. Contrariwise, if you configure **protocol-filter exclusive**, then bridge is executed for exclusive in the configured filters. This is default.

You cannot add the enabled routing protocols to the router (protocols displayed on running **configuration** from the monitoring menu) 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 OUI**L**P (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:

Filtering for NetBIOS SAP (DSAP FO) packets entering the bridge through port 1.

```
ASRT config>protocol-filter dsap FO 1
ASRT config>
```

Example 2:

Filtering for Ethernet XNS (0600) packets entering the bridge through port 2.

```
ASRT config>protocol-filter ether 0600 2
ASRT config>
```

Example 3:

Filtering for SNAP AppleTalk Phase 2 (08-00-07-80-9B) packets entering the bridge through port 3.

```
ASRT config>protocol-filter 080007809B 3
ASRT config>
```

Example 4:

Filtering for all Ethernet packets except IP and ARP (0800 and 0806) entering the bridge through port 1.

```
ASRT config>protocol-filter inclusive
ASRT config>protocol-filter ether 0800 1
ASRT config>protocol-filter ether 0806 1
ASRT config>
```

9.2.21 ROUTE-PROTOCOL

Enables routing for a protocol.

Syntax:

```
ASRT config>route-protocol ?
  ip    IP protocol group
  ipv6    IPv6 protocol group
```

Example:

Run **route-protocol ip** to enable routable packet routing for the group of IPv4 protocols.

```
ASRT config>route-protocol ip
ASRT config>
```

Run **route-protocol ipv6** to enable routable packet routing for the IPv6 protocol group.

```
ASRT config>route-protocol ipv6
ASRT config>
```

9.2.22 SET

Run the **set** command to set the following parameters:

- Aging time for dynamic address entries in the filtering database.
- Bridge MAC address.
- Size of the bridge filtering database.
- 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.

Syntax:

```
ASRT config>set ?
  age                       Time for aging out dynamic entries
  bridge                    Sets the bridge address
  filtering                 Entries that can be held in the filtering database
  lf-bit-interpretation     Largest Frame (LF) bit encoding interpretation
  maximum-packet-size       Largest MAC service data unit (MSDU) size
  port                      Enables or disables a bridge port
  protocol                  Modifies the spanning tree or port parameters
  route-descriptor-limit    Associate a maximum RD length for ARE or STE frames
ASRT config>
```

9.2.22.1 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 with SR-TB bridge characteristics.

Syntax:

```
ASRT config>set age <age-time> <resolution>
```

age-time	Dynamic entries aging timer. The aging timer default value is 300 seconds. This age timer can take values between 10 and 1,000,000 seconds.
resolution	Resolution used to check the dynamic entries timeout. Dynamic entry timeout checking is carried out using the time period indicated in this parameter. The resolution default value is 5 seconds, permitting a range between 1 and 60 seconds.

Example:

Setting a dynamic entry age time of 250 seconds with timeout checking every 20 seconds.

```
ASRT config>set age 250 20
ASRT config>
```

9.2.22.2 SET BRIDGE

Sets the bridge MAC address. Use this command when the configured bridge does not participate in any interface with a MAC address (e.g., serial line bridge).

Syntax:

```
ASRT config>set bridge <MAC-address>
```

Example:

```
ASRT config>set bridge 001122334455
ASRT config>
```

Note

Each bridge in the network must have a unique MAC address so the spanning tree protocol works properly.

This is the low order 6-octet bridge address in the bridge identifier. Where a MAC address is not configured in the bridge, the device uses the MAC of the lowest numbered port with the associated MAC address, if this exists.

To delete the configured MAC, so the device automatically selects the MAC, configure a MAC address containing all zeros.

Example:

```
ASRT config>set bridge 00-00-00-00-00-00
ASRT config>
```

9.2.22.3 SET FILTERING

Sets the number of entries that can be held in the bridge filtering database. If you don't configure the size of the database, a table with 1024 entries for each bridge port is created by default. The size of the filtering database can be checked by running list filtering.

Syntax:

```
ASRT config>set filtering <size>
```

Example:

```
ASRT config>set filtering 1024
ASRT config>
```

9.2.22.4 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       Only three bits of the routing control field are used
  extended    Six bits of the routing control field are used
```

9.2.22.4.1 SET LF-BIT-INTERPRETATION BASIC

In *basic* mode only three bits of the routing control field are used. *Extended* and *basic* modes are compatible.

Example:

```
ASRT config>set lf-bit-interpretation basic
ASRT config>
```

9.2.22.4.2 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. Default is *extended*. *Extended* and *basic* modes are compatible.

Example:

```
ASRT config>set lf-bit-interpretation extended
ASRT config>
```

9.2.22.5 SET MAXIMUM-PACKET-SIZE

Sets the largest MAC Service Data Unit (MSDU) size for a port, if source routing is enabled on said 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 packet size configured for the interface associated with the port.

Syntax:

```
ASRT config>set maximum-packet-size <Port Number> < MSDU size>
```

Example:

```
ASRT config>set maximum-packet-size 2 4000
MSDU is adjusted to 2052
ASRT config>
```

9.2.22.6 SET PORT

Enables or disables a bridge port.

Syntax:

```
ASRT config>set port ?
  disable     Disables a port for those with bridge configured
  enable      Enables a port for those having bridge configured
```

9.2.22.6.1 SET PORT DISABLE

Disables a bridge port. The port status passes to disabled.

Example:

```
ASRT config>set port disable 2
ASRT config>
```

9.2.22.6.2 SET PORT ENABLE

Disables a bridge port.

Example:

```
ASRT config>set port enable 2
ASRT config>
```

9.2.22.7 SET ROUTE-DESCRIPTOR-LIMIT

Configures the maximum number of Route Descriptors (RD) the Routing Information Field (RIF) can contain in both the All Route Explorer (ARE) frames and Spanning Tree Explorer (STE) frames. These frames are forwarded by the bridge where source routing is enabled. In other words, this configures the maximum number of hops for the ARE and STE frames.

Syntax:

```
ASRT config>set route-descriptor-limit <are | ste> <hop-count>
  are    Associate a maximum RD length for ARE frames
  ste    Associate a maximum RD length for STE frames
```

hop-count	Maximum number of permitted hops for the specified type of frame. The configurable values are from 0 to 255. Default is 14.

Example 1:

Configuring the maximum number of permitted hops for STE frames to 10 hops.

```
ASRT config>set route-descriptor-limit ste 10
ASRT config>
```

Example 2:

Configuring the maximum number of permitted hops for ARE frames to 20 hops.

```
ASRT config>set route-descriptor-limit are 20
ASRT config>
```

9.2.22.8 SET SPANNING-TREE

Configures the various Spanning Tree protocol parameters. when the PVST mode is enabled, that parameters configure the VLAN 1 (default), to configure the VLANs from 2 to 4096, use the *set spanning-tree vlan* command which has the same options.

Syntax:

```
ASRT config>set spanning-tree ?
  bpdu-filter         Configures BPDU filter globally
  bpdu-guard          Configures BPDU guard globally
  bridge-priority     Configures Bridge Priority
  forward-delay       Configures Bridge Forward Delay
  max-age             Configures Bridge Max Age
  port                Spanning tree protocol port parameters
  protocol-version    Configures Spanning-Tree Protocol Version
  tx-hold-count       Configures Transmit Hold Count
  vlan                Configure the pvst parameters of a VLAN
```

9.2.22.8.1 SET SPANNING-TREE BPDU-FILTER DEFAULT

Enables BPDU filtering by default in all ports in an *Edge Port* state. To disable default BPDU filtering, run the **no** command.

Syntax:

```
ASRT config>set spanning-tree bpdu-filter default
```

9.2.22.8.2 SET SPANNING-TREE BPDU-GUARD DEFAULT

Enables BPDU guard by default on all ports in an *Edge Port* state. To disable default BPDU guard, run the **no** command.

Syntax:

```
ASRT config>set spanning-tree bpdu-guard default
```

9.2.22.8.3 SET SPANNING-TREE BRIDGE-PRIORITY

Configures the priority assigned to bridge, together with the bridge MAC address, used to form the bridge identifier. Configure a value between 0 to 65535. The 12 least significant priority bits should be 0. In cases where they aren't, the device will round up to the nearest valid priority. The priority default for bridge is 32768.

Syntax:

```
ASRT config>set spanning-tree bridge-priority <Bridge-Priority>
```

Example:

```
ASRT config>set spanning-tree bridge-priority 32120
value rounded to 28672
ASRT config>
```

9.2.22.8.4 SET SPANNING-TREE FORWARD-DELAY

Configures the time interval waited before changing the state in a port (where bridge is selected as Spanning Tree bridge root). Configure a value between 4 to 30 seconds. Default is 15.

Syntax:

```
ASRT config> set spanning-tree forward-delay <forward-delay>
```

Example:

```
ASRT config>set spanning-tree forward-delay 4
ASRT config>
```

Note

When setting this parameter, make sure that the following condition is met:

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

9.2.22.8.5 SET SPANNING-TREE MAX-AGE

Configures the maximum duration (time period) the information received in the spanning tree protocol is valid. Configure a value between 6 and 40 seconds. Default is 20.

Syntax:

```
ASRT config>set spanning-tree max-age <max-age>
```

Example:

```
ASRT config>set spanning-tree max-age 13
ASRT config>
```

Note

When setting this parameter, make sure that the following conditions are met:

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

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

The Bridge Hello Time value is not configurable. This is set to 2 seconds.

9.2.22.8.6 SET SPANNING-TREE PORT

Configures the Spanning Tree protocol parameters for a bridge port.

Syntax:

```
ASRT config>set protocol port <port-number>
  bpdu-filter  don't send or receive BPDUs for this port
       enable     Enables BPDU filtering on this port
       disable    Disables BPDU filtering on this port
  bpdu-guard   don't accept BPDUs on this port
       enable     Enables BPDU guard on this port
       disable    Disables BPDU guard on this port
  edge-port    Edge Port configuration
        admin    Configures the port as an edge port
        auto     Configures auto edge port detection
  enable         enables spanning-tree on this port
  disable        disables spanning-tree on this port
  link-type    Link Type Configuration
        point-to-point         Configures the port as conected to a point-to-point
                               LAN
        point-to-multipoint    Configures the port as not conected to a
                               point-to-point LAN
        autodetect             Automatic detection of a point-to-point LAN
  path-cost    Port Path Cost
        <0..200000000>    Port Path-cost
  priority     Port Priority
        <0..255>    Port Priority
```

Port Number	Bridge port number: selects the port the Spanning Tree parameters are configured for.
Bpdu-filter	Enables (**bpdu-filter enable** option) or disables (**bpdu-filter disable** option) BPDU filtering in the port.
Bpdu-guard	Enables (**bpdu-guard enable** option) or disables (**bpdu-guard disable** option) BPDU guard in the port.
Edge port	Configures the operating parameters for the RSTP bridge detection state machine. If a port is established as being directly connected to a station (*Edge Port* port), the Spanning Tree protocol convergence in this port is faster.The **edge-port admin** option configures the port as *Edge Port* by default. However the state machine can pass the port to a *Non Edge Port* when it receives BPDUs through said port. If you select the **edge-port auto** option, the Spanning Tree protocol decides if this port is directly connected to a station or not. Both options are not exclusive. By default the port is not detected as *Edge Port* and the autodetection doesn't come into operation.
enable	Enables the Spanning Tree protocol on a specific bridge port.
disable	Disables the Spanning Tree protocol on a specific bridge port.

Note

Deactivating the Spanning Tree protocol in a port may produce loops in the network due to parallel bridges.

Link-type	Configures the port link-type: connected to a point-to-point link (**link-type point-to-point** option), connection to a point-to-multipoint link (**link-type point-to-multipoint** option), or detected by the Spanning Tree protocol (**link-type autodetect** option). The latter is the default option.
Path Cost	Cost associated with the port used in the Spanning Tree protocol for possible root path cost. The range is 1 to 65535. 0 indicates the default path cost. Cost is automatically assigned depending on the type of interface the port refers to.
Port Priority	Port priority. This is between 0 and 255. The 4 least significant priority bits should be 0. Where they aren't, the device will round up to the nearest valid priority. The priority default value is 128.

Example 1:

Configuration for cost associated with port 2.

```
ASRT config>set spanning-tree port 2 10000
ASRT config>
```

Example 2:

Configuration for priority associated with port 3.

```
ASRT config> set spanning-tree port 3 priority 56
value rounded to 48
ASRT config>
```

Example 3:

Configuration for port 1 as a port directly connected to a station (a link not shared with other bridges).

```
ASRT config>set spanning-tree port 1 edge-port admin
ASRT config>
```

Example 4:

Link configuration, which port 2 is connected to, as a point-to-point link (port is directly connected to another bridge).

```
ASRT config>set spanning-tree port 2 link-type point-to-point
ASRT config>
```

Example 5:

Configuring BPDU filtering in port 3.

```
ASRT config>set spanning-tree port 3 bpdu-filter enable
ASRT config>
```

9.2.22.8.7 SET SPANNING-TREE PROTOCOL-VERSION

Configures the Spanning Tree protocol being run in the device.

Syntax:

```
ASRT config>set spanning-tree protocol-version
  stp-compatibility        STP Compatibility Mode
  rstp-normal-operation    RSTP Normal Operation
```

stp-compatibility	The device is running the old Spanning Tree protocol, defined in the 802.1D-1998. Spanning Tree BPDUs are used and RSTP (Rapid Spanning Tree Protocol) frames are not sent.
rstp-normal-operation	The device is running the Rapid Spanning Tree protocol. This protocol is an upgrade of the Spanning Tree for quicker convergence. This is default.

Example:

```
ASRT config>set spanning-tree protocol-version stp-compatibility
ASRT config>
```

9.2.22.8.8 SET SPANNING-TREE TX-HOLD-COUNT

Maximum number of BPDUs that can be sent through a port per second. Configure a value between 1 and 10 seconds. Default is 6 BPDUs per second.

Syntax:

```
ASRT config>set spanning-tree tx-hold-count <tx-hold-count>
```

Example:

```
ASRT config>set spanning-tree tx-hold-count 2
ASRT config>
```

9.2.22.8.9 SET SPANNING-TREE VLAN

In the PVST mode, configure the previously cited parameters but for the specified VLAN, from 2 to 4094.

Syntax:

```
ASRT config>set spanning-tree vlan <2..4094> ?
  bpdu-filter         Configures BPDU filter globally
  bpdu-guard          Configures BPDU guard globally
  bridge-priority     Configures Bridge Priority
  forward-delay       Configures Bridge Forward Delay
  max-age             Configures Bridge Max Age
  port                Spanning tree protocol port parameters
  protocol-version    Configures Spanning-Tree Protocol Version
  tx-hold-count       Configures Transmit Hold Count
```

Command history:

Release	Modification
11.01.09	The command and its options were introduced.

9.2.23 SOURCE-ROUTING

Enables source routing for a given port. Use this command when source routing is required on part of the bridge. If source routing is the only feature required, disable transparent bridging on the interface.

Syntax:

```
ASRT config>source-routing <port> <segment>
```

port	Bridge port.
segment	12-bit number representing the LAN/WAN the port is connected to. All the ports for other bridges connected 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 connected to this LAN/WAN have the same perspective of the LAN/WAN identification value.

Note

If two segments already configured (i.e., a 1:N SRB configuration), create a virtual segment before enabling source routing in a new port.

Example:

```
ASRT config>source-routing 2 3
ASRT config>
```

9.2.24 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.

Syntax:

```
ASRT config>spanning-tree-explorer <port>
```

Example:

```
ASRT config>spanning-tree-explorer 1
ASRT config>
```

9.2.25 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 on source routing stations from the RIFs from 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.

For the conversion to work properly, a segment number must be assigned to the transparent bridging domain. SR-TB bridges connected to this domain should be configured with the same segment number.

Syntax:

```
ASRT config>sr-tb-conversion <TB-segment> <TB-MTU>
```

TB-segment	Number of the transparent domain segment (Transparent Bridge, TB). The configurable range is from 1 to FFF hexadecimal.
TB-MTU	Transparent domain MTU (Transparent Bridge, TB). The configurable range is from 1 to 65535.

Example:

```
ASRT config>sr-tb-conversion 2 1400
TB-Domain's MTU is adjusted to 1350
ASRT config>
```

9.2.26 STP

Globally enables the Spanning Tree protocol.

Syntax:

```
ASRT config>stp
```

9.2.27 TRANSPARENT

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

Syntax:

```
ASRT config>transparent <port>
```

Example:

```
ASRT config>transparent 2
ASRT config>
```

9.2.28 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.

Syntax:

```
ASRT config>ub-encapsulation
```

9.2.29 VIRTUAL-BRIDGE

Accesses the virtual-bridge entity configuration. Where an entity doesn't exist, it will be created. A virtual bridge is an entity independent of the bridge, where you can associate device interfaces. The same device interface cannot be configured in various bridge entities. You can create up to 8 bridge entities, including the main entity, which are configurable from the ASRT menu with identifier 0. The virtual bridge feature allows you to work with a single physical device, which behaves as several independent bridges.

Run **virtual-bridge** to access the virtual bridge configuration menu required (VBDG menu). The virtual entity menu basically consists of the same commands as the main entity menu (ASRT menu), with the exception of the options relative to BAN and DLS, which are only operative in the main entity.

Syntax:

```
ASRT config>virtual-bridge <entity>
```

entity	Virtual bridge identifier, this can take values between 1 and 7.

Example:

```
ASRT config>virtual-bridge 2

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

9.2.30 VIRTUAL-SEGMENT

Sets the virtual segment number used in the SRB 1:N bridge.

Syntax

```
ASRT config>virtual-segment <num-segment>
```

num-segment	Bridge virtual segment identification number. This can take values between 1 and FFF hexadecimal.

Example:

```
ASRT config>virtual-segment 2
```

9.2.31 VLAN

Accesses the virtual LAN configuration menu (VLANs). For further information on the VLAN configuration, please see manual bintec *Dm751-I VLAN*.

Syntax:

```
ASRT config>vlan

802.1Q Bridge Configuration
ASRT VLAN Config>
```

9.2.32 EXIT

Exits the bridge configuration menu and returns to the main configuration menu.

Syntax:

```
ASRT config>exit
```

Example:

```
ASRT config>exit
Config>
```

Chapter 10 Bridge Monitoring

10.1 Accessing bridge monitoring

This section describes the bridge monitoring commands.

To access the main bridge entity monitoring menu, run **protocol asrt** (main monitoring menu).

```
+protocol asrt

ASRT+
```

Note

The bridge must be enabled to access the bridge monitoring.

From the main entity monitoring menu, ASRT menu, access the monitoring menu for a virtual entity by running **virtual-bridge**.

```
+protocol asrt
ASRT+virtual-bridge 1
ASRT Virtual Bridge 1+
```

Note

Precreate virtual bridge entity monitoring (in the configuration menu) in order to access it.

10.2 Bridge monitoring commands

This section describes the bridge monitoring commands.

10.2.1 ? (HELP)

Displays the commands available in the current menu. After a specific command, this displays the available options.

Syntax:

```
ASRT+?
  list              Lists the available bridges that can be configured
  virtual-bridge    Accesses the selected bridge monitoring prompt
  exit
ASRT+
```

10.2.2 LIST

Displays the configured bridge entities.

Example:

```
ASRT+list
Number  Bridge ID (prio/add)    Status
======  ====================== ======
0       32768/00-a0-26-70-01-dc Enabled
1       32768/00-a0-26-00-03-33 Disabled
ASRT+
```

Number	Bridge entity identifier. Identifier 0 corresponds to the main bridge entity.
Bridge ID	Bridge identifier, composed of its priority and MAC address.
Status	Bridge status: active or not.

10.2.3 VIRTUAL-BRIDGE

Accesses the monitoring menu for a virtual bridge entity.

Syntax:

```
ASRT+virtual-bridge <id>
```

An identifier equal to 0 is used to refer to the main bridge entity. The monitoring commands are the same for all the bridge entities. In Section 3, Bridge Entity Monitoring Commands, the bridge entity monitoring commands are explained.

The prompt displayed in the monitoring menu is different depending on whether we are dealing with the main entity (ASRT Main Bridge+) or a virtual entity (ASRT Virtual Bridge 1+).

10.2.4 EXIT

Exits the bridge monitoring menu.

Example:

```
ASRT+exit
+
```

10.3 Monitoring commands for a bridge entity

Command	Function
? (HELP)	Displays the available commands.
ADD	Adds static entries to the filter database.
BAN	Accesses the BAN monitoring menu (Boundary Access Node).
CLEAR	Clears the bridge statistics.
CACHE	Displays cache entries for a specified port.
DELETE	Deletes an entry in the filter database.
FLIP-MAC-ADDRESS	Flips MAC address from canonical to 802.5 (non-canonical or IBM) bit order and vice versa.
LIST	Displays information on the bridge configuration and functionality.
NAME-CACHING	Accesses the Name Caching facility monitoring menu and NetBIOS duplicated frame filtering.
NETBIOS	Accesses the NetBIOS monitoring menu.
SPANNING-TREE	Configures Spanning Tree protocol parameters.
EXIT	Exits the bridge entity monitoring menu.

10.3.1 ? (HELP)

Displays the commands available in the current menu. After a specific command, the available options are displayed.

Syntax:

```
ASRT Main Bridge+?
```

Example:

```
ASRT Main Bridge+?
  add                 Adds static entries to the filtering database
  ban                 Access to the BAN monitoring prompt
  cache               Lists entries from the cache for a specific port
  clear               Clears bridge information
  delete              Deletes an entry from the filtering database
  flip-mac-address    Converts a MAC address from a canonical format to 802.5
                      format and vice versa
  list                Lists configuration and operation information about
                      bridging
  name-caching        Access to the Name Caching monitoring prompt
  netbios             Access to the NetBIOS monitoring prompt
```

```
  spanning-tree       Spanning Tree configuration functionality
  exit
ASRT Main Bridge+
```

10.3.2 ADD

Adds static entries to the filter database. These entries are not permanent: if you reboot the device, they are deleted.

Syntax:

```
ASRT Main Bridge+add ?
  destination-address-filter    Adds a destination address filter entry to the
                                filtering database
  static-entry                  Adds an static entry to the filtering database
```

10.3.2.1 ADD DESTINATION-ADDRESS-FILTER

Adds a destination address filter entry to the bridge filter database.

Example:

Filtering of frames with destination MAC address equal to 00-01-02-03-04-05.

```
ASRT Main Bridge+add destination-address-filter ?
  <mac>    Destination MAC address
ASRT Main Bridge+add destination-address-filter 00-01-02-03-04-05
ASRT MAuin Bridge+
```

10.3.2.2 ADD STATIC-ENTRY

Adds a static address entry to the bridge filter permanent database. As well as the MAC address associated with the entry, specify the port mapping you want. This allows you, for a MAC address, to indicate what ports can be used as output for each input port.

The output ports are optional.

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

Example:

Creation of a static entry associated with MAC address 11-22-33-44-55-66, so those packets with this destination address (which enter through port 1), are not sent through any port and packets entering through port 2 are only sent through port 3.

```
ASRT Main Bridge+add static-entry ?
  <mac>    MAC address
ASRT Main Bridge+add static-entry 11-22-33-44-55-66 ?
  <0..254>    Input port number (0 for any)
ASRT Main Bridge+add static-entry 11-22-33-44-55-66 1 ?
  <0..254>    Output port number, (0 for none)
ASRT Main Bridge+add static-entry 11-22-33-44-55-66 1 0
ASRT Main Bridge+
ASRT Main Bridge+add static-entry ?
  <mac>    MAC address
ASRT Main Bridge+add static-entry 11-22-33-44-55-66 ?
  <0..254>    Input port number (0 for any)
ASRT Main Bridge+add static-entry 11-22-33-44-55-66 2 ?
  <0..254>    Output port number, (0 for none)
ASRT Main Bridge+add static-entry 11-22-33-44-55-66 2 3 ?
  <0..254>    Output port, (0 to end)
  <cr>
ASRT Main Bridge+add static-entry 11-22-33-44-55-66 2 3 0
ASRT Main Bridge+
```

10.3.3 BAN

Accesses the BAN monitoring menu. For further information on the BAN monitoring, please see the manual **bintec** *Dm716-I DLSw Protocol* .

Syntax:

```
ASRT Main Bridge+ban
```

Example:

```
ASRT Main Bridge+ban
Boundary Access Node Console
BAN>
```

10.3.4 CACHE

Displays the contents of a selected bridging port routing cache. If the port does not have a cache, the following message is displayed:

```
PORT DOESNT HAVE A CACHE
```

Syntax:

```
ASRT Main Bridge+cache ?
<0..254>    Port number
```

Example:

```
ASRT Main Bridge+cache 2
MAC Address    MC*  Entry Type       Age  Port(s)
00-00-93-00-c0-d0   Dynamic           20  2 (token-ring3/0)
ASRT Main Bridge+
```

MAC Address	6-byte MAC address of the entry.
Entry Type	Displays one of the following address entry types: *Dynamic:* Addresses learned by the bridge dynamically. These entries time out if, after a specified time, they are not refreshed. They are also deleted should the device be switched off and on. *Unknown:* Unknown to the bridge. May be bugs and/or illegal addresses.
Age	Age in seconds for a dynamic entry. Age is decremented at each resolution interval so, when it reaches zero, the entry becomes invalid.
Port(s)	The port number associated with the entry. Displays the interface name (the name of the interface with the cache).

10.3.5 CLEAR

Clears the bridge statistics.

Syntax:

```
ASRT Main Bridge+clear ?
  spanning-tree-protocol    Lists information related to the Spanning Tree
                            protocol
```

10.3.5.1 CLEAR SPANNING-TREE-PROTOCOL

Clears the Spanning Tree protocol statistics.

Syntax:

```
ASRT Main Bridge+clear spanning-tree-protocol ?
  counters    Clears Spanning Tree protocol counters
```

COUNTERS. Clears counters for the BPDUs transmitted and received by the Spanning Tree protocol.

Syntax:

```
ASRT Main Bridge+clear spanning-tree-protocol counters ?
  vid    VLAN identifier
```

Syntax:

```
ASRT Main Bridge+clear spanning-tree-protocol counters vid ?
  <2..4094>    Value in the specified range
```

```
  all           All VLAN instances
  default       Default VLAN instance
```

Example:

```
ASRT Main Bridge+clear spanning-tree-protocol counters vid default
```

Command history:

Release	Modification
11.01.09	The vid option was introduced.

10.3.6 DELETE

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

Syntax:

```
ASRT Main Bridge+delete ?
  <mac>    MAC address
```

Example:

Deletes the static entry associated with address 11-22-33-44-55-66.

```
ASRT Main Bridge+delete 11-22-33-44-55-66
```

10.3.7 FLIP-MAC-ADDRESS

Displays specific MAC addresses in the canonical and non-canonical formats by flipping the address bit order. The command interprets a MAC address entered, with or without separation hyphens, as a MAC address in canonical format, and a MAC address entered, with a colon to separate it, as a non-canonical format address.

Syntax:

```
ASRT Main Bridge+flip-mac-address ?
  <mac>    MAC address
```

Example 1:

You want to find out the non-canonical format for MAC address 00-a0-26-44-22-11.

```
ASRT Main Bridge+flip-mac-address 00-a0-26-44-22-11
IEEE 802 canonical bit order:    00-a0-26-44-22-11
IBM Token-Ring native bit order: 00:05:64:22:44:88

ASRT Main Bridge+
```

Or entered without separation hyphens:

```
ASRT Main Bridge+flip-mac-address 000a26442211
IEEE 802 canonical bit order:    00-0a-26-44-22-11
IBM Token-Ring native bit order: 00:50:64:22:44:88

ASRT Main Bridge+
```

Example 2:

You want to find out the canonical format for MAC address 00:05:64:22:44:89.

```
ASRT Main Bridge+flip-mac-address 00:05:64:22:44:89
IEEE 802 canonical bit order:    00-a0-26-44-22-91
IBM Token-Ring native bit order: 00:05:64:22:44:89

ASRT Main Bridge+
```

10.3.8 LIST

Displays information on the bridge configuration and functionality.

Syntax:

```
ASRT Main Bridge+list ?
  adaptive                  Lists all the general information related to SR-TB
                            bridge
  bridge                    Lists all the general information related to the
                            bridge
  conversion                Lists the conversion rules from functional
                            addresses to group addresses
  database                  Lists the contents of the filtering database used
                            in transparent bridging
  filtering                 Lists the contents of the database used for the
                            protocol filtering
  port                      Lists port states
  source-routing            Lists configuration information of source routing
                            bridge
  spanning-tree-protocol    Lists information related to Spanning Tree protocol
  transparent               Lists configuration information about transparent
                            bridging
```

10.3.8.1 LIST ADAPTIVE

Lists all general information on the SR-TB bridge, which converts between types of bridging.

Syntax:

```
ASRT Main Bridge+list adaptive ?
  config      Lists general configuration information related to SR-TB bridge
  counters    Lists SR-TB bridge statistics
  database    Lists the elements of the RIF database used by the SR-TB bridge
```

Config	Displays general configuration information on the SR-TB bridge.
Counters	Displays the SR-TB bridge counters.
Database	Displays contents of the SR-TB bridge RIF database.

10.3.8.1.1 LIST ADAPTIVE CONFIG

Example:

```
ASRT Main Bridge+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
   2  001      token-ring3/0    Enabled    2052     Yes  Yes
   -  001      Adaptive         Enabled    1470
ASRT Main Bridge+
```

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 exceeding said time limit are discarded.
Aging granularity	How often entries are scanned to look for expiries complying with the aging timer.
Port	Number of a port associated with conversion bridging. This only displays source routing ports.
Segment	Source-routing segment number assigned to the port.
Interface	Interface associated with the bridge port.
State	Current state of the bridge port.
MTU	Maximum frame size (from the end of the RIF to the beginning of the FCS) the bridge can manage in a specified segment.
DUP: TSF STE	Indicates if duplicated STE (Spanning Tree Explorer) or TSF (Transparent Spanning Frames) frames are sent.

10.3.8.1.2 LIST ADAPTIVE COUNTERS

Syntax:

```
ASRT Main Bridge+list adaptive counters
Hash collision count:                0
Adaptive database overflow count:    0
ASRT Main Bridge+
```

Hash Collision Count	Number of addresses stored (hashed) in 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 an address was overwritten as the conversion database table ran out of table space.

10.3.8.1.3 LIST ADAPTIVE DATABASE

Displays certain portions of the adaptive bridge RIF database. This is due to the potential size of the database.

Syntax:

```
ASRT Main Bridge+list adaptive database ?
  address         Lists information from the database related to a MAC address
  all-segments    Lists all information from the database
  port            Lists all entries from the bridge database for a specific
                  port
  segment         Lists all entries from the bridge database for a specific
                  segment
```

Address	Displays data on the address found in the database.
All-segments	Displays the entire database.
Port	Displays all bridge entries in the database for a specific port.
Segment	Displays all bridge entries in the database associated with a specified segment number.

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

Syntax 1:

```
ASRT Main Bridge+list adaptive database address ?
  <mac>    MAC address
```

Example 1:

```
ASRT Main Bridge+list adaptive database address 00a026400ba4
Canonical MAC address:           00-a0-26-40-0b-a4
IBM Token-Ring native address:   80:05:64:02:d0:25
Via port:                        1 (token-ring3/0)
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 Main Bridge+
```

Example 2:

```
ASRT Main Bridge+list adaptive database all-segments

Canonical Address  Interface       Port  Seg    Age RIF: Type  Direct  Len    LF
IBM MAC Address    RIF

00-00-93-78-b7-3a  token-ring3/0      1  100    310       ARE Reverse    6  1470
80:00:c9:1e:ed:5c  869010012000

00-a0-26-40-0b-a4  token-ring3/0      1  100    320       ARE Reverse    6  1470
80:05:64:02:d0:25  869010012000

ASRT Main Bridge+
```

Syntax 3:

```
ASRT Main Bridge+list adaptive database port ?
  <0..254>    Port number
```

Example 3:

```
ASRT Main Bridge+list adaptive database port 2
Canonical Address  Interface       Port  Seg    Age RIF: Type  Direct  Len    LF
IBM MAC Address    RIF

00-0a-83-78-b7-a4  token-ring3/0      2  200    300       ARE Reverse    6  1470
80:00:c9:1e:ed:25  869010011000

ASRT Main Bridge+
```

Syntax 4:

```
ASRT Main Bridge+list adaptive database segment ?
  <hex 1..fff>    Segment number
```

Example 4:

```
ASRT Main Bridge+list adaptive database segment 100
Canonical Address  Interface       Port  Seg    Age RIF: Type  Direct  Len    LF
IBM MAC Address    RIF

00-00-93-78-b7-3a  token-ring3/0      1  100    315       ARE Reverse   6   1470
80:00:c9:1e:ed:5c  869010012000

00-a0-26-40-0b-a4  token-ring3/0      1  100    320       ARE Reverse   6   1470
80:05:64:02:d0:25  869010012000

ASRT Main Bridge+
```

The following information is displayed for each entry:

Canonical address	MAC address for the node corresponding to this entry, displayed in the canonical format.
Interface	Identifier for the network interface that learned this entry.
Port	Identifier for the port that learned this entry.
Seg	Identifier for the segment that learned this entry.
Age	Entry age in seconds. When this reaches zero, it's deleted.
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.
IBM MAC Address	MAC address for the node corresponding to this entry, displayed in IBM non-canonical format.
RIF	RIF (Routing Information Field) learned from this node.

10.3.8.2 LIST BRIDGE

Lists general information about the bridge for the STP instances.

Syntax:

```
ASRT Main Bridge+list bridge ?
  vid    VLAN identifier
  <cr>    Default VLAN instance
```

Syntax:

```
ASRT Main Bridge+list bridge vid ?
  <2..4094>    Value in the specified range
  all          All VLAN instances
  default      Default VLAN instance
```

Example:

```
ASRT Main Bridge+list bridge vid default
```

```
Bridge Vid              1
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
Filtering age:         320 seconds
Filtering resolution:  5 seconds

                                                    Maximum
Port  Interface        State    MAC Address          Modes  MSDU  Segment  Flags
   1  ethernet0/0      Up       00-a0-26-40-0c-e4    SR     2096     100     RD
   2  ethernet0/0.1    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 Main Bridge+
```

Bridge Vid	VLAN identifier of the STP instance.
Bridge ID (prio/add)	Bridge identifier.
Bridge State	Indicates whether bridging is enabled or disabled.
UB-Encapsulation	Indicates if 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 said bridge.
STP Participation	Participation type in the Spanning Tree Protocol.
Filtering age	Lifetime associated with the database dynamic entries.
Filtering resolution	Resolution used when checking the expiry for dynamic entries in the database.
Port	Port identifier.
Interface	Interface associated with the port.
State	The current state of the interface (Up or Down).
MAC address	The MAC address associated with said port in canonical bit order.
Modes	The bridging mode for the port. *T* indicates transparent bridging. *SR* indicates source routing.
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 said 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.

Command history:

Release	Modification
11.01.09	The vid option was introduced.

10.3.8.3 LIST CONVERSION

Displays the rules to convert functional addresses to group addresses used by the bridge, depending on the type of frame.

Syntax:

```
ASRT Main Bridge+list conversion ?
  all          Lists all conversion rules related to all types of frames
  ethertype    Lists all conversion rules related to Ethernet frames
```

```
  sap          Lists all conversion rules related to SAP frames
  snap         Lists all conversion rules related to SNAP frames
```

10.3.8.3.1 LIST CONVERSION ALL

Displays the conversion rules associated with all types of frames.

Example:

```
ASRT Main Bridge+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 Main Bridge+
```

10.3.8.3.2 LIST CONVERSION ETHERTYPE

Displays the conversion rules associated with Ethernet frames. It's possible to indicate a specific Ethernet type or display the rules associated with all Ethernet types.

Syntax:

```
ASRT Main Bridge+list conversion ethertype ?
  <hex 0..ffff>    Ethernet type in hexadecimal (0 for all)
```

Example:

```
ASRT Main Bridge+list conversion ethertype 0
Ethernet type 0800 translations:
Group 11-22-33-44-55-66 <=> Functional 40-cc-44-88-44-cc (02:33:22:11:22:33)
```

10.3.8.3.3 LIST CONVERSION SAP

Displays the conversion rules associated with SAP frames. It's possible to indicate a specific SAP type or display the rules associated with all SAP 802.2 types.

Syntax:

```
ASRT Main Bridge+list conversion sap ?
  <hex 0..100>    SAP in hexadecimal (100 for all)
```

Example:

```
ASRT Main Bridge+list conversion sap 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 Main Bridge+
```

10.3.8.3.4 LIST CONVERSION SNAP

Displays the conversion rules associated with SNAP frames. It's possible to indicate a specific SNAP type or display the rules associated with all SNAP 802.2 types.

Syntax:

```
ASRT Main Bridge+list conversion snap ?
  <10 hex chars>    SNAP protocol identifier (0000000000 for all)
```

Example:

```
ASRT Main Bridge+list conversion snap 0000006002
IEEE 802 SNAP PID 00-00-00-60-02 translations:
Group ab-00-00-02-00-00 <=> Functional 03-00-04-00-00-00 (c0:00:20:00:00:00)
```

10.3.8.4 LIST DATABASE

Lists the contents of transparent filtering databases.

Syntax:

```
ASRT Main Bridge+list database ?
  all-ports    Lists all the content of the database used for transparent
               bridging
  dynamic      Lists all the dynamics entries (learned) from the address
               database
  local        Lists all the local entries (reserved) from the address database
  permanent    Lists all the permanents entries from the address database
  port         Lists all the entries from the address database related to a
               specific port
  range        Lists an entries range from the address database
  static       Lists all the static entries from the address database
```

10.3.8.4.1 LIST DATABASE ALL-PORTS

Displays the entire transparent bridging database.

Example:

```
ASRT Main Bridge+list database all-ports
MAC Address    MC*   FID  VID Entry Type       Age  Port(s)

00-00-0c-07-ac-08      1    1 Dynamic          320  1 (ethernet0/0)
00-00-e8-3d-26-97      1    1 Dynamic          295  1 (ethernet0/0)
00-00-e8-3d-a5-04      1    1 Dynamic          320  1 (ethernet0/0)
00-01-02-21-14-e0      1    1 Dynamic          170  1 (ethernet0/0)
00-01-02-21-1b-12      1    1 Dynamic          305  1 (ethernet0/0)
00-01-02-ae-a6-e6      1    1 Dynamic          275  1 (ethernet0/0)
00-01-02-dc-ca-a3      1    1 Dynamic          315  1 (ethernet0/0)
00-01-02-dc-ca-a5      1    1 Dynamic          320  1 (ethernet0/0)
00-01-02-dc-cb-65      1    1 Dynamic           35  1 (ethernet0/0)
00-a0-24-51-cb-9b      1    1 Dynamic          215  1 (ethernet0/0)
00-a0-24-7c-ec-fd      1    1 Dynamic          125  1 (ethernet0/0)
00-a0-26-00-01-a8      1    1 Dynamic          255  1 (ethernet0/0)
00-a0-26-00-5e-10      1    1 Dynamic          320  1 (ethernet0/0)
00-a0-26-32-26-d8      1    1 Dynamic          320  1 (ethernet0/0)
00-a0-26-32-c4-70      1    1 Dynamic           50  1 (ethernet0/0)
00-a0-26-32-c5-68      1    1 Dynamic          295  1 (ethernet0/0)
00-a0-26-44-03-38      1    1 Registered            1 (ethernet0/0)
00-a0-26-44-16-b8      1    1 Dynamic          315  1 (ethernet0/0)
00-a0-26-44-1c-d8      1    1 Dynamic           45  1 (ethernet0/0)
00-a0-26-5c-5f-aa      1    1 Dynamic          100  1 (ethernet0/0)
00-a0-26-60-00-24      1    1 Dynamic          310  1 (ethernet0/0)
00-c0-9f-fe-33-5a      1    1 Dynamic           95  1 (ethernet0/0)
00-d0-b7-a0-03-1e      1    1 Dynamic           20  1 (ethernet0/0)
00-d0-e9-40-31-d6      1    1 Dynamic          315  1 (ethernet0/0)
00-e0-63-11-e3-39      1    1 Dynamic          115  1 (ethernet0/0)
01-80-c2-00-00-00*     1    1 Registered            1
01-80-c2-00-00-01*     1    1 Reserved              All
01-80-c2-00-00-02*     1    1 Reserved              All
01-80-c2-00-00-03*     1    1 Reserved              All
01-80-c2-00-00-04*     1    1 Reserved              All
01-80-c2-00-00-05*     1    1 Reserved              All
01-80-c2-00-00-06*     1    1 Reserved              All
01-80-c2-00-00-07*     1    1 Reserved              All
01-80-c2-00-00-08*     1    1 Reserved              All
01-80-c2-00-00-09*     1    1 Reserved              All
01-80-c2-00-00-0a*     1    1 Reserved              All
01-80-c2-00-00-0b*     1    1 Reserved              All
01-80-c2-00-00-0c*     1    1 Reserved              All
01-80-c2-00-00-0d*     1    1 Reserved              All
01-80-c2-00-00-0e*     1    1 Reserved              All
01-80-c2-00-00-0f*     1    1 Reserved              All
```

```
02-00-01-11-00-02      1    1 Dynamic          300  1 (ethernet0/0)
02-00-01-11-00-03      1    1 Dynamic          320  1 (ethernet0/0)
02-00-01-11-00-05      1    1 Dynamic          305  1 (ethernet0/0)
02-00-01-11-00-07      1    1 Dynamic          310  1 (ethernet0/0)
02-0a-00-01-66-01      1    1 Dynamic          320  1 (ethernet0/0)
03-00-00-00-80-00*     1    1 Reserved              All
08-00-20-83-56-ff      1    1 Dynamic          270  1 (ethernet0/0)
08-00-5a-93-6d-fa      1    1 Dynamic          315  1 (ethernet0/0)

ASRT Main Bridge+
```

Note

The fields described below are displayed for all the **list database** command options.

MAC Address	MAC address in canonical format.	
*MC**	An asterisk following an address entry indicates the entry has been flagged as a multicast address.	
FID	Filtering Identifier. Used to define common filtering/forwarding behavior for a group of VLANs.	
VID	VLAN Identifier. When a packet is received without a tag, it is classified as belonging to the default VLAN associated with the input port.	
Entry Type	Specifies one of the following types:	
	Reserved	Address reserved by the IEEE802.1D standard.
	Registered	Addresses internally registered by the bridge itself so it works correctly.
	Permanent	Entries permanently created in the configuration process. These entries are not deleted in cases of power on/offs.
	Static	Static entries creating in the monitoring processes. These entries are ageless but are deleted in power on/offs.
	Dynamic	Dynamic entries learned by the bridge. These entries time out if they are not refreshed after a certain time and are deleted in power on/offs.
	Free	Free entries in the database not associated with any MAC address. This type is not used and should not be seen except in exceptional conditions, where a conflict between the bridge updating the database and its viewing through the monitoring process.
	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. When the age zeroizes, it is deleted.	
Port(s)	The port identifier for that entry. For dynamic entries, the port the entry has been learned through is indicated. Interface type is also listed for single port entries.	

10.3.8.4.2 LIST DATABASE DYNAMIC

Displays all dynamic (learned) address database entries.

Example:

```
ASRT Main Bridge+list database dynamic
MAC Address    MC*   FID  VID Entry Type       Age  Port(s)

00-00-0c-07-ac-08      1    1 Dynamic          320  1 (ethernet0/0)
00-00-e8-3d-a5-04      1    1 Dynamic          320  1 (ethernet0/0)
00-01-02-dc-ca-a5      1    1 Dynamic          265  1 (ethernet0/0)
00-01-02-dc-cb-aa      1    1 Dynamic          270  1 (ethernet0/0)
00-01-03-ba-5d-14      1    1 Dynamic          315  1 (ethernet0/0)
00-01-03-ba-82-74      1    1 Dynamic          105  1 (ethernet0/0)
00-01-03-ba-82-97      1    1 Dynamic          260  1 (ethernet0/0)
```

```
00-01-03-ba-82-c1      1    1 Dynamic           45  1 (ethernet0/0)
00-01-6c-3c-45-b2      1    1 Dynamic          260  1 (ethernet0/0)
ASRT Main Bridge+
```

10.3.8.4.3 LIST DATABASE LOCAL

Displays all local (reserved) address database entries.

Example:

```
ASRT Main Bridge+list database local
MAC Address    MC*   FID  VID Entry Type       Age  Port(s)

00-a0-26-40-0c-e4      1    1 Registered            1 (token-ring3/0)
00-a0-26-40-0c-e5      1    1 Registered            2 (ethernet0/0)
01-80-c2-00-00-00*     1    1 Registered            1
ASRT Main Bridge+
```

10.3.8.4.4 LIST DATABASE PERMANENT

Displays all permanent address database entries.

Example:

```
ASRT Main Bridge+list database permanent
MAC Address    MC*   FID  VID Entry Type       Age  Port(s)

00-11-22-33-44-55      1    1 Permanent             1 (token-ring3/0)  ->  1-2
ASRT Main Bridge+
```

10.3.8.4.5 LIST DATABASE PORT

All entries in the address database associated with a particular port are displayed.

Syntax:

```
ASRT Main Bridge+list database port ?
  <0..254>    Port number
```

Example:

```
ASRT Main Bridge+list database port 1
MAC Address    MC*   FID  VID Entry Type       Age  Port(s)

00-00-0c-07-ac-08      1    1 Dynamic          320  1 (ethernet0/0)
00-00-e8-3d-a5-04      1    1 Dynamic          320  1 (ethernet0/0)
00-01-02-dc-ca-a5      1    1 Dynamic          265  1 (ethernet0/0)
00-01-02-dc-cb-aa      1    1 Dynamic          270  1 (ethernet0/0)
00-01-03-ba-5d-14      1    1 Dynamic          315  1 (ethernet0/0)
00-01-03-ba-82-74      1    1 Dynamic          105  1 (ethernet0/0)
00-01-03-ba-82-97      1    1 Dynamic          260  1 (ethernet0/0)
00-01-03-ba-82-c1      1    1 Dynamic           45  1 (ethernet0/0)
00-01-6c-3c-45-b2      1    1 Dynamic          260  1 (ethernet0/0)

ASRT Main Bridge+
```

10.3.8.4.6 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 within this range are displayed.

Syntax:

```
ASRT Main Bridge+list database range ?
  <mac>    First MAC address
ASRT Main Bridge+list database range 00-00-00-00-00-00 ?
  <mac>    Last MAC address
ASRT Main Bridge+
```

Example:

```
ASRT Main Bridge+list database range 00-00-00-00-00-00 FF-FF-FF-FF-FF-FF
```

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

00-00-0c-07-ac-08      1    1 Dynamic          320  1 (ethernet0/0)
00-00-e8-3d-a5-04      1    1 Dynamic          320  1 (ethernet0/0)
00-01-02-dc-ca-a5      1    1 Dynamic          265  1 (ethernet0/0)
00-01-02-dc-cb-aa      1    1 Dynamic          270  1 (ethernet0/0)
00-01-03-ba-5d-14      1    1 Dynamic          315  1 (ethernet0/0)
00-01-03-ba-82-74      1    1 Dynamic          105  1 (ethernet0/0)
00-01-03-ba-82-97      1    1 Dynamic          260  1 (ethernet0/0)
00-01-03-ba-82-c1      1    1 Dynamic           45  1 (ethernet0/0)
00-01-6c-3c-45-b2      1    1 Dynamic          260  1 (ethernet0/0)

ASRT Main Bridge+
```

10.3.8.4.7 LIST DATABASE STATIC

Displays static entries from the address database.

Example:

```
ASRT Main Bridge+list database static
MAC Address    MC*   FID  VID Entry Type       Age  Port(s)

01-02-03-0a-0b-0c*     1    1 Static               1 (ethernet0/0)  ->  1-2
ASRT Main Bridge+
```

10.3.8.5 LIST FILTERING

Displays the database content used for protocol filtering.

Syntax:

```
ASRT Main Bridge+list filtering ?
  all          Lists all content from the filtering protocol database
  ethertype    Lists entries from the filtering database related to Ethernet
               protocol
  sap          Lists entries from the filtering database related to SAP
               protocol
  snap         Lists entries from the filtering database related to SNAP
               protocol
```

10.3.8.5.1 LIST FILTERING ALL

Displays all filtering database entries per protocol.

Example:

```
ASRT Main Bridge+list filtering all
Ethernet type 9000 is bridged & routed on ports 1-2
IEEE 802.2 destination SAP 00 is bridged & routed 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 & routed on ports 1-2
ASRT Main Bridge+
```

Descriptors, used to explain how packets are processed, include the following:

- *Routed* - Packets are passed to the routing forwarder to be forwarded.
- *Filtered* - Packets are administratively filtered by user setting protocol filters.
- *Bridged and routed* - The packets are passed to the routing forwarder to be processed. Additionally, they are processed by the bridge, which decides what ports the packets should be sent through.

All descriptors described above also apply to ARP packets with this Ethertype.

10.3.8.5.2 LIST FILTERING ETHERTYPE

Displays Ethernet protocol type filter database entries.

Syntax:

```
ASRT Main Bridge+list filtering ethertype ?
```

```
  <hex 0..ffff>    Ethernet type in hexadecimal (0 for all)
```

Example:

```
ASRT Main Bridge+list filtering ethertype 0
Ethernet type 0800 is no bridged & routed on ports 1
Ethernet type 888e is no bridged & routed on ports 1
Ethernet type 88c7 is bridged & routed on ports 1
Ethernet type 9000 is bridged & routed on ports 1
ASRT Main Bridge+
```

10.3.8.5.3 LIST FILTERING SAP

Displays SAP protocol filter database entries.

Syntax:

```
ASRT Main Bridge+list filtering sap ?
  <hex 0..100>    SAP in hexadecimal (100 for all)
```

Example:

```
ASRT Main Bridge+list filtering sap 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 Main Bridge+
```

10.3.8.5.4 LIST FILTERING SNAP

Displays SNAP protocol identifier filter database entries.

Syntax:

```
ASRT Main Bridge+list filtering snap ?
  <10 hex chars>    SNAP protocol identifier (0000000000 for all)
```

Example:

```
ASRT Main Bridge+list filtering snap 0000000000
IEEE 802 SNAP PID 00-00-00-90-00 is bridged & processed on ports 1-2
ASRT Main Bridge+
```

10.3.8.6 LIST PORT

Displays the status of the bridge ports for the STP instances.

Syntax:

```
ASRT Main Bridge+list port ?
  <-1..254>    Port number (-1 for all)
```

Syntax:

```
ASRT Main Bridge+list port -1 ?
  vid    VLAN identifier
  <cr>    Default VLAN instance
```

Syntax:

```
ASRT Main Bridge+list port -1 vid ?
  <2..4094>    Value in the specified range
  all          All VLAN instances
  default      Default VLAN instance
```

Example:

```
ASRT Main Bridge+list port -1 vid default
VLAN of the port : 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: token-ring3/0
------------------------------------------------------------------------------
VLAN of the port : 1
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: ethernet0/0
------------------------------------------------------------------------------
ASRT Main Bridge+
```

VLAN of the port	VLAN of the STP instance to which the port belongs.
Port ID (dec)	Port Identifier. This displays priority associated with the port and the port number, both in decimal and hexadecimal.
Port State	Port state: *Blocking*, if the received packets are dropped, *Listening*, if the received packets are dropped but are in transition towards processing them, *Learning*, if the received packets are not processed, but used to learn MAC addresses, *Forwarding*, if the received packets are processed and *Undefined*, if not in any of the above states.
STP Participation	Shows whether or not the port participates in the Spanning Tree protocol.
Port Supports	Indicates if the port is configured to operate in the transparent bridge domain, source routing or both.
SRB	Information associated with the source routing bridge. Indicates the segment number, MTU and if bridging is carried out over STE frames.
Duplicates Frames Allowed	Information associated with the transparent bridge. Indicates if duplicated STE and ARE frames are allowed.
Assoc Interface Name	Interface identifier associated with the port. For Frame-Relay ports, this also displays the circuit name.

Command history:

Release	Modification
11.01.09	The vid option was introduced.

10.3.8.7 LIST SOURCE ROUTING

Displays **source-routing** bridge configuration information.

Syntax:

```
ASRT Main Bridge+list source-routing ?
  configuration    Lists general information related to SRB bridge
  counters         Lists SRB bridge statistics
  state            Lists information related to the state of the SRB bridge
```

10.3.8.7.1 LIST SOURCE-ROUTING CONFIGURATION

Displays general information on the SRB bridge of the STP instances.

Syntax:

```
ASRT Main Bridge+list source-routing configuration ?
  vid    VLAN identifier
  <cr>    Default VLAN instance
```

Syntax:

```
ASRT Main Bridge+list source-routing configuration vid ?
  <2..4094>    Value in the specified range
  all          All VLAN instances
  default      Default VLAN instance
```

Example:

```
ASRT Main Bridge+list source-routing configuration vid default
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
  2        001    token-ring3/0     Enabled    2052    Auto
  -        001    Adaptive          Enabled    1470    Yes
ASRT Main Bridge+
```

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	Port identifier.
Segment	Assigned segment number for the network connected to this port.
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 leave through this port.

Command history:

Release	Modification
11.01.09	The vid option was introduced.

10.3.8.7.2 LIST SOURCE-ROUTING COUNTERS

Displays all SRB bridge counters.

Syntax:

```
ASRT Main Bridge+list source-routing counters ?
  all-ports    Lists statistics for all ports
  port         Lists statistics for a specific port
  segment      Lists statistics for a specific segment
```

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 Main Bridge+list source-routing counters all
Counters for port 1, segment 100, interface token-ring3/0:
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 Main Bridge+
```

Port	Port identifier.
Segment	Segment identifier in hexadecimal.
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) sent as Transparent Bridge frames.
TB Frames Sent as SR	Transparent bridge frames (received on this interface) sent as source routing frames.
Dropped, in queue overflow	Frames dropped: input queue overflow.
Dropped, source address filter	Frames dropped: source address matches a source address filter in the filtering database.
Dropped, destination address filter	Frames dropped: destination address matches a source address filter in the filtering database.
Dropped, protocol filtering	Frames dropped: protocol identifier is being administratively filtered.
Dropped, invalid ri length	Frames dropped: RIF length is less than 2 or over 30.
Dropped, duplicate segment	Frames dropped: duplicate segment in the RIF. This is normal for ARE frames.
Dropped, segment mismatch	Frames dropped: the outgoing segment number does not match any in this bridge.
Dropped, duplicated lan id	Frames dropped: duplicated LAN ID.
Dropped, stehop count exceeded	Frames dropped: the STE has surpassed the number of permitted hops.
Dropped, arehop count exceeded	Frames dropped: the ARE has surpassed the number of permitted hops.
Dropped, no buffer available	Frames dropped: no buffer available.
Dropped, mtu exceeded	Frames dropped: the MTU has been exceeded.

Syntax 2:

```
ASRT Main Bridge+list source-routing counters port ?
  <0..254>    Port number
```

Example 2:

```
ASRT Main Bridge+list source-routing counters port 1
Counters for port 1, segment 100, interface token-ring3/0:
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 Main Bridge+
```

Syntax 3:

```
ASRT Main Bridge+list source-routing counters segment ?
  <hex 1..fff>    Segment number
```

Example 3:

```
ASRT Main Bridge+list source-routing counters segment 100
Counters for port 1, segment 100, interface token-ring3/0:
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 Main Bridge+
```

10.3.8.7.3 LIST SOURCE-ROUTING STATE

Displays information on the SRB bridge status of the STP instances.

Syntax:

```
ASRT Main Bridge+list source-routing state ?
  vid    VLAN identifier
  <cr>    Default VLAN instance
```

Syntax:

```
ASRT Main Bridge+list source-routing state vid ?
  <2..4094>    Value in the specified range
  all          All VLAN instances
  default      Default VLAN instance
```

Example:

```
ASRT Main Bridge+list source-routing state vid default

Bridge state:               Up

Port   Segment    Interface         State  STE Forwarding
   2       001    token-ring3/0     Up          Yes
ASRT Main Bridge+
```

Command history:

Release	Modification
11.01.09	The vid option was introduced.

10.3.8.8 LIST SPANNING-TREE-PROTOCOL

Displays spanning tree protocol information. The transparent bridge uses the spanning tree protocol to form a loop-free topology.

Syntax:

```
ASRT Main Bridge+list spanning-tree-protocol ?
  configuration    Lists configuration information about Spanning Tree protocol
  counters         Lists counters related to Spanning Tree protocol
  detail           Lists detailed information about operation of Spanning Tree
                   protocol
  state            Lists the state of the Spanning Tree protocol
  tree             Lists current information about Spanning Tree protocol
```

10.3.8.8.1 LIST SPANNING-TREE-PROTOCOL CONFIGURATION

Displays information on the spanning tree protocol.

Syntax:

```
ASRT Main Bridge+list spanning-tree-protocol configuration ?
  vid    VLAN identifier
  <cr>    Default VLAN instance
```

Syntax:

```
ASRT Main Bridge+list spanning-tree-protocol configuration vid ?
  <2..4094>    Value in the specified range
  all          All VLAN instances
  default      Default VLAN instance
```

Example:

```
ASRT Main Bridge+list spanning-tree-protocol configuration vid default
Bridge ID (prio/add):  28672/00-a0-26-44-03-38
VlanID:                1
Maximum age:           20.000 seconds
Hello time:            2.000 seconds
Forward delay:         15.000 seconds
Transmit Hold Count:   6
Migrate Time:          3 seconds

Port  Interface      Priority          Cost      State
   1  ethernet0/0         128       2000000      Enabled
   2  ethernet0/1         128        200000      Enabled
ASRT Main Bridge+
```

Bridge ID (prio/add)	Bridge Identifier. Displays the bridge priority and MAC address.
VlanID	VLAN identifier of the STP instance.
Maximum age	Value of the maximum age parameter used by the Spanning Tree protocol. Indicates the maximum time the information received in a BPDU is valid for.
Hello Time	Value of the *Hello Time* parameter used by the Spanning Tree protocol. Indicates Hello BPDUs periodic sending.
Forward Delay	Value of the *Forward Delay* parameter used by the Spanning Tree protocol. Indicates the wait time in a *Learning* state before passing to the *Forwarding* state.
Transmit Hold Count	Value of the *Transmit Hold Count* parameter used by the Spanning Tree protocol. Indicates the maximum number of BPDUs that can be sent through a port in one second.
Migrate Time	Value of the *Migrate Time* parameter used by the Spanning Tree protocol. This parameter is used to start the timers verifying if Rapid Spanning Tree BPDUs should be used, or Spanning Tree BPDUs, and if a port can be considered as a directly connected port.
Port	Port identifier.
Interface	Interface associated with the port.
Priority	Port priority.
Cost	Cost associated with the port in the Spanning Tree.
State	Port state: active or not.

Command history:

Release	Modification
11.01.09	The vid option and the VlanId field were introduced.

10.3.8.8.2 LIST SPANNING-TREE-PROTOCOL COUNTERS

Displays spanning tree protocol counters.

Syntax:

```
ASRT Main Bridge+list spanning-tree-protocol counters ?
  vid    VLAN identifier
```

```
  <cr>    Default VLAN instance
```

Syntax:

```
ASRT Main Bridge+list spanning-tree-protocol counters vid ?
  <2..4094>    Value in the specified range
  all          All VLAN instances
  default      Default VLAN instance
```

Example:

```
ASRT Main Bridge+list spanning-tree-protocol counters vid default
VLAN Id:                                  1
BPDUs received:                           0
        Dropped:                          0
        Errs:                             0
        Config:                           0
        TCN:                              0
        RST:                              0
BPDUs sent:                               30
        Dropped:                          0
        Errs:                             0
        Config:                           0
        TCN:                              0
        RST:                              30

Port  Interface                           BPDUs rcv
----  ----------------  ------------------------------------------------------
                            Total     Drop      Err       TCN     Conf       RST
   1  ethernet0/0                0        0        0         0        0         0
   2  ethernet0/1                0        0        0         0        0         0

Port  Interface                           BPDUs xmt
----  ----------------  ------------------------------------------------------
                            Total        Err        TCN        Conf        RST
   1  ethernet0/0               15          0          0          0          15
   2  ethernet0/1               15          0          0          0          15

Port  Interface         Forward transitions
----  ----------------  -------------------
   1  ethernet0/0       1
   2  ethernet0/1       1
ASRT Main Bridge+
```

VlanID	VLAN identifier of the STP instance.
BPDUs received	Number of protocol (BPDU) frames received. This shows the total number of BPDUs, both globally and through the interface. Additionally, the following is itemized: BPDUs dropped (*Dropped*), BPDUs received with errors (*Errs*), configuration BPDUs (*Config*), BPDUs notifying a change in topology (*TCN*) and BPDUs for the Rapid Spanning Tree protocol (*RST*).
BPDUs sent	Number of protocol (BPDU) frames sent. This shows the total number of BPDUs, both globally and through the interface. Additionally, the following is itemized: BPDUs sent with errors (*Errs*), configuration BPDUs (*Config*), BPDUs notifying a change in topology (*TCN*) and BPDUs for the Rapid Spanning Tree protocol (*RST*).
Forward transitions	Number of times the port has passed to a *Forwarding* state.

Command history:

Release	Modification
11.01.09	The vid option and the VlanId field were introduced.

10.3.8.8.3 LIST SPANNING-TREE-PROTOCOL DETAIL

Displays detailed information on spanning tree protocol operations for each STP instance. This shows the state of all internal variables used during spanning tree operations.

Syntax:

```
ASRT Main Bridge+list spanning-tree-protocol detail ?
  all       Lists all the information about the Spanning Tree protocol
  bridge    Lists information about Spanning Tree protocol related to the
            bridge
  port      Lists information about Spanning Tree protocol related to a
            specific port
```

all	Displays all the information on the spanning tree.
bridge	Displays the operating information on the spanning tree, globally associated with the bridge.
port	Displays the operating information on the spanning tree, associated with a particular port.

In the bridge and port options there is an option to display the different STP instances selecting the VLAN:

Syntax:

```
ASRT Main Bridge+list spanning-tree-protocol detail bridge ?
  vid    VLAN identifier
  <cr>    Default VLAN instance
```

Syntax:

```
ASRT Main Bridge+list spanning-tree-protocol detail bridge vid ?
  <2..4094>    Value in the specified range
  all          All VLAN instances
  default      Default VLAN instance
```

Example:

```
ASRT Main Bridge+list spanning-tree-protocol detail all
                ----------------
                Brige Parameters
                ----------------
PVST VID ......................... 1
Bridge Id ........................ 32768/00-a0-26-44-03-38
rstpBEGIN ........................ FALSE
rstp_sched ....................... FALSE
Bridge Message Age ............... 0.000
Bridge Max Age ................... 20.000
Bridge Hello Time ................ 2.000
Bridge Forward Delay ............. 15.000
Transmit Hold Count .............. 6
Force Protocol Version .......... 2 (RSTP Normal Operation)
BPDU filtering ................... enabled by default
BPDU guard ....................... enabled by default
Root priority vector:
        RootBridgeID ............. 32768/00-a0-26-44-03-38
        RootPathCost ............. 0
        DesignatedBridgeId ...... 32768/00-a0-26-44-03-38
        DesignatedPortID ........ 0 (0/0)
        BridgePortID ............. 0 (0/0)
Root times:
        Message Age .............. 0.000
        Max Age .................. 20.000
        Hello Time ............... 2.000
        Forward Delay ............ 15.000
State Machines:
        Bridge role selection ... ROLE_SELECTION

                ---------------
                Port Parameters
                ---------------
Port 1
        Port priority ..................... 128
        MAC Operational ................... Yes
        Administrative state .............. Enabled
        AuthControlledPortStatus .......... Authorized
        Operational Point To Point MAC .... Not Point To Point
        Admin Point To Point MAC .......... Auto
```

```
Port enabled ..................... Yes
BPDU filtering ................... enabled
BPDU guard ....................... disabled (by default)
Port path cost ................... 2000000
Oper Edge ........................ Non Edge
Rcv BPDU ......................... No
Rcv RSTP ......................... No
Rcv STP .......................... Yes
Rcv msg .......................... No
Send RSTP ........................ No
Rcv info ......................... No
mcheck ........................... No
newInfo .......................... No
Tx Count ......................... 0
role ............................. Designated
selectedRole ..................... Designated
infoIs ........................... Mine
learn ............................ Yes
learning ......................... Yes
forward .......................... Yes
forwarding ....................... Yes
sync ............................. No
synced ........................... No
proposing ........................ No
proposed ......................... No
agree ............................ No
agreed ........................... No
disputed ......................... No
reselect ......................... No
selected ......................... Yes
updtInfo ......................... No
reRoot ........................... No
fdbFlush ......................... No
tcAck ............................ No
rcvdTc ........................... No
rcvdTcn .......................... No
rcvdTcAck ........................ No
tcProp ........................... No
AdminEdge ........................ No
AutoEdge ......................... No
Ageing Time ...................... 320
rapid Ageing ..................... No
Port priority vector:
        RootBridgeID ............. 32768/00-a0-26-44-03-38
        RootPathCost ............. 0
        DesignatedBridgeId ....... 32768/00-a0-26-44-03-38
        DesignatedPortID ......... 32769 (128/1)
        BridgePortID ............. 32769 (128/1)
Port times:
        Message Age .............. 0.000
        Max Age .................. 20.000
        Hello Time ............... 2.000
        Forward Delay ............ 15.000
Designated priority vector:
        RootBridgeID ............. 32768/00-a0-26-44-03-38
        RootPathCost ............. 0
        DesignatedBridgeId ....... 32768/00-a0-26-44-03-38
        DesignatedPortID ......... 32769 (128/1)
        BridgePortID ............. 0 (0/0)
Designated times:
        Message Age .............. 0.000
        Max Age .................. 20.000
        Hello Time ............... 2.000
        Forward Delay ............ 15.000
Message priority vector:
        RootBridgeID ............. 32768/00-17-0e-82-e6-c2
        RootPathCost ............. 0
```

```
                DesignatedBridgeId ........ 32768/00-17-0e-82-e6-c2
                DesignatedPortID .......... 32769 (128/1)
                BridgePortID .............. 32769 (128/1)
        Message times:
                Message Age ............... 0.000
                Max Age ................... 20.000
                Hello Time ................ 2.000
                Forward Delay ............. 15.000
        Timers:
                edgeDelayWhile ............ 0
                fdWhile ................... 0
                helloWhen ................. 1
                mdelayWhile ............... 0
                rbWhile ................... 0
                rcvdInfoWhile ............. 0
                rrWhile ................... 0
                tcWhile ................... 0
        Machine State Status:
                Receive State Machine ..... RECEIVE
                Receive State Machine ..... RECEIVE
                Transmit State Machine ..... IDLE
                Protocol Migration State Machine ... SENSING
                Bridge Detection State Machine ..... NOT_EDGE
                Port Information State Machine ..... CURRENT
                Role Transitions State Machine ..... DESIGNATED_PORT
                State Transition State Machine ..... FORWARDING
                Topology Change State Machine ...... ACTIVE
more ? n

ASRT Main Bridge+
```

Command history:

Release	Modification
11.01.09	The vid option and the PVST VID field were introduced.

10.3.8.8.4 LIST SPANNING-TREE-PROTOCOL STATE

Displays information on the current state of the spanning tree protocol for each STP instance.

Syntax:

```
ASRT Main Bridge+list spanning-tree-protocol state ?
  vid     VLAN identifier
  <cr>    Default VLAN instance
```

Syntax:

```
ASRT Main Bridge+list spanning-tree-protocol state vid ?
  <2..4094>    Value in the specified range
  all          All VLAN instances
  default      Default VLAN instance
```

Example:

```
ASRT Main Bridge+list spanning-tree-protocol state
Designated root (prio/add):    32768/00-a0-26-44-03-38
VlanID:                        1
Root cost:                     0
Root port:                     1 (ethernet0/0)
Current (root) Maximum Age:    20.000 seconds
Current (root) Hello Time:     2.000 seconds
Current (root) Forward Delay:  15.000 seconds

Port      Interface         State         Role
   1      ethernet0/0       Forwarding    Designated
   2      ethernet0/1       Forwarding    Designated
ASRT Main Bridge+
```

Designated root	Identifier of the bridge selected as root bridge for the Spanning Tree protocol.

VlanID	VLAN identifier of the STP instance.
Root cost	Cost associated with the path to the root bridge.
Root port	Identifier for the port selected as root port on this bridge. Where the bridge has been selected as root bridge, *Self* is shown, indicating there is no root port.
Current maximum age	Value of *Max Age* parameter indicated by the root bridge.
Current hello time	Value of *Hello Time* parameter indicted by the root bridge.
Current Forward Delay	Value of *Forward Delay* parameter indicated by the root bridge.
Port	Port identifier.
Interface	Interface associated with the port.
State	State of the port for the Spanning Tree: *Discarding*, if the received packets are dropped, *Listening*, if the received packets are not processed but are used to learn MAC addresses, *Forwarding*, if the received packets are processed and *Undefined*, if not in any of the above states.
Role	Port role in the Spanning Tree protocol. The role can be *Disabled, Designated, Root, Alternate* or *Backup*.

Command history:

Release	Modification
11.01.09	The vid option and the VlanID field were introduced.

10.3.8.8.5 LIST SPANNING-TREE-PROTOCOL TREE

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

Syntax:

```
ASRT Main Bridge+list spanning-tree-protocol tree ?
  vid     VLAN identifier
  <cr>    Default VLAN instance
```

Syntax:

```
ASRT Main Bridge+list spanning-tree-protocol tree vid ?
  <2..4094>    Value in the specified range
  all          All VLAN instances
  default      Default VLAN instance
```

Example:

```
ASRT Main Bridge+list spanning-tree-protocol tree
Port                              Designated  Desig.             Designated Des.
 N. Interface                           Root   Cost                  Bridge Port
 1 token-ring3/0    32768/00-a0-26-40-0c-e4      0 32768/00-a0-26-40-0c-e4 80-01
 2 ethernet0/0      32768/00-a0-26-40-0c-e4      0 32768/00-a0-26-40-0c-e4 80-02
ASRT Main Bridge+
```

Port	Port identifier.
Interface	Interface associated with the port.
Designated root	Root bridge identifier sent by the bridge authorized for the LAN where this port is connected.
Designated cost	Cost associated with the path to the root bridge by the port authorized for the LAN, which said port is connected to.
Designated Bridge	Bridge identifier authorized for the LAN which said port is connected to.
Designated Port	Port identifier authorized for the LAN, which said port is connected to.

Command history:

Release	Modification
11.01.09	The vid option was introduced.

10.3.8.9 LIST TRANSPARENT

Displays transparent bridge configuration information.

Syntax:

```
ASRT Main Bridge+list transparent ?
  configuration    Lists general information related to transparent bridging
  counters         Lists transparent bridging statistics
  state            Lists status information about transparent bridging
```

10.3.8.9.1 LIST TRANSPARENT CONFIGURATION

Displays information on transparent bridge.

Example:

```
ASRT Main Bridge+list transparent configuration
Filtering database size:    2066
Aging time:                 320 seconds
Aging granularity           5 seconds

Port    Interface      State        MTU
   2    ethernet0/0    Enabled      1514
ASRT Main Bridge+
```

Filtering database size	Size of the database used for filtering.
Aging Time	Lifetime of the dynamic entries in the database.
Aging granularity	Resolution used in the database dynamic entries timeout checking.
Port	Port identifier.
Interface	Interface associated with the port.
State	Port state: active or not.
MTU	Maximum size of the frame that can be sent or received through the port.

10.3.8.9.2 LIST TRANSPARENT COUNTERS

Displays the transparent bridge counters.

Syntax:

```
ASRT Main Bridge+list transparent counters ?
  all-ports    List statistics from all ports
  port         Lists statistics from a specific port
```

All-Ports	Displays the counters for all the bridge ports.
Port	Displays the counters for one particular port.

Example:

```
ASRT Main Bridge+list transparent counters port 2
Counters for port 2, interface ethernet0/0:
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
Dropped, malformed frames:                        0
Frames sent by bridging:                        423
Dropped, dest port blocked:                       0
Dropped, transmit error:                          0
Dropped, too big to send on port:                 0

ASRT Main Bridge+
```

Total frames received by interface	Total frames received in the port.
Frames submitted to bridging	Frames managed by the bridge.
Frames submitted to routing	Frames managed by the router.
Dropped, source address filtering	Frames dropped: source address filtering.
Dropped, dest address filtering	Frames dropped: destination address filtering.
Dropped, protocol filtering	Frames dropped: protocol filtering.

Dropped, no buffer available to copy	Frames dropped: lack of buffers.
Dropped, input queue overflow	Frames dropped: lack of space in the input queue.
Dropped, source port blocked	Frames dropped: source port is blocked.
Dropped, malformed frames	Frames dropped: they are malformed.
Frames sent by bridging	Frames sent by the bridge through this port.
Dropped, dest port blocked	Frames dropped: destination port is blocked.
Dropped, transmit error	Frames dropped: transmission errors.
Dropped, too big to send on port	Frames dropped: too big.

Command history:

Release	Modification
10.08.36.01.04, 10.08.43, 10.09.08.01.15, 10.09.21, 11.00.00.02.06, 11.00.03	Malformed frames counter introduced.

10.3.8.9.3 LIST TRANSPARENT STATE

Displays the transparent bridge state information.

Example:

```
ASRT Main Bridge+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 Main Bridge+
```

Filtering database size	Size of the database used for filtering.
Number of static entries	Number of static entries in the database.
Number of dynamic entries	Number of dynamic entries in the database.
Hash collision count	Number of addresses that were stored (through the hash function) in the same location in the hash table. This number is accumulative and reflects the total number of hash collisions that have occurred. Increases in this number can indicate a possible problem in the size of the table.
Filtering database overflow	Number of times that a database entry has been overwritten due to lack of space.

10.3.9 NAME-CACHING

Accesses the Name Caching facility monitoring menu and the duplicated frame filtering for NetBIOS.

Syntax:

```
ASRT Main Bridge+name-caching



Name Cache+
```

Commands	Function
? (HELP)	Displays all the monitoring commands, or lists options for specific commands.
LIST	Displays all statistics and counters related to Name Caching and duplicated frames filtering for NetBIOS.
PORT	Selects a specific port for monitoring purposes.
EXIT	Exits the name caching and duplicated frames filtering monitoring menu.

10.3.9.1 ? (HELP)

Use the **?** (HELP) command to list the available commands. If this is introduced after a command, you can list the available options.

Example:

```
Name Cache+?
  list    Displays name caching and duplicated frames filtering information
  port    Accesses to the name caching menu for a specific port
  exit
Name Cache+
```

10.3.9.2 LIST

Displays the current statistics and counters for the Name Caching and duplicated frames filtering. This information can be displayed on a global or a per interface basis by running the **port** monitoring command.

Syntax:

```
Name Cache+list ?
  add-names    Displays duplicated frames filtering database
  cache        Displays name caching information
```

10.3.9.2.1 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+
```

Name	Device identifier name.
MAC	Device MAC address.
Add (Group) Name Received	Counter for received Add-Name and Add-Group-Name frames.
Add (Group) Name Filtered	Counter for filtered Add-Name and Add-Group-Name frames.

10.3.9.2.2 LIST CACHE

Syntax:

```
Name Cache+list cache ?
  rifs          Displays name caching database
  statistics    Displays name caching 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+
```

10.3.9.3 PORT

Accesses the name cache and the duplicated frame filtering monitoring submenu for a specified port.

Example:

```
Name Cache+port 2
Name Cache Port+
```

The following commands are available within the port submenu:

Syntax:

```
Name Cache Port+?
  list    Displays name caching and duplicated frames filtering information
  exit
```

10.3.9.3.1 LIST

Syntax:

```
Name Cache Port+list ?
  add-names    Displays duplicated frames filtering statistics
  cache        Displays name caching statistics
```

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 aggregated 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+
```

10.3.9.3.2 EXIT

Exits the monitoring menu for a specific port, returning to the name cache and NetBIOS frame filtering global monitoring menu.

Example:

```
Name Cache Port+exit
Name Cache+
```

10.3.9.4 EXIT

Exits the name cache and NetBIOS frame filtering monitoring menu, returning to the bridge monitoring menu.

Example:

```
Name Cache+exit
ASRT Main Bridge+
```

10.3.10 NETBIOS

Accesses the NetBIOS monitoring menu.

See *NetBIOS Filtering and Caching commands* on page 105, for an explanation of the NetBIOS monitoring commands.

Syntax:

```
ASRT Main Bridge+netbios
```

Note

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

NetBIOS Support not in load.

10.3.11 SPANNING TREE

Options associated with the Spanning Tree protocol.

Syntax:

```
ASRT Main Bridge+spanning-tree
  force-bpdu-migration-check    Forces BPDU migration check
```

SPANNING-TREE FORCE-BPDU-MIGRATION-CHECK

Forces RSTP BPDU frame sending in the port specified during migration time. You can check to see that there are no STP bridges in the LAN and can send RSTP BPDU frames in the port.

Syntax:

```
ASRT Main Bridge+spanning-tree force-bpdu-migration-check <port-number>
```

Example:

```
ASRT Main Bridge+spanning-tree force-bpdu-migration-check 1
```

10.3.12 EXIT

Exits the virtual bridge entity monitoring menu and returns to the bridge monitoring global menu.

Syntax:

```
ASRT Main Bridge+exit
```

Example:

```
ASRT Main Bridge+exit
ASRT+
```

Chapter 11 Using NetBIOS

11.1 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 I-frame traffic sent on established LLC2 connections.

The bintec router allows you to define NetBIOS configuration parameters that are different and independent for each of the virtual bridges enabled on the device. In the same way, this maintains all the caches and state memories associated with the NetBIOS operations independently for each bridge.

11.1.1 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.

11.1.2 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.

11.1.3 NetBIOS sessions setup procedure

To establish a NetBIOS session for data transfer 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.

How to establish a session:

(1) The client repeatedly broadcasts a Spanning Tree Explorer (STE) NetBIOS UI frame containing the server NetBIOS name 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, which contains the server MAC address and, for Token Ring, the route to the server.
The client can then do either of the following:
   - Establish an LLC 2 connection to communicate with the server using I-frames.
   - Begin to communicate with the server using specifically-routed NetBIOS UI frames.

11.2 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 through either the source or destination MAC address.
Frame type	Specific types of NetBIOS frames.
Duplicate frame	Duplicate frames.
Response	Responses to 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, the name caching and route caching control how the router forwards the remaining frames.

Using MAC Filtering on page 146 describes MAC address filtering.

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

11.2.1 Frame type filtering

Frame type filtering filters the following types of frames:

- Name Conflict Resolution.
- General Broadcast.
- Trace Control.

Name Conflict Resolution Frame Filtering

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's essential that the NetBIOS names for the devices, used to establish a NetBIOS session (normally a server), are unique.
- It tends to be important that the individual NetBIOS names for the devices, within the same group (or domain), are also unique.
- It is often *not* particularly important that the NetBIOS names for the devices, used to establish NetBIOS sessions (typically client), are unique, particularly over domains.

  This means that networks, where the server names are well controlled, may find use of name conflict resolution frame filtering advantageous. This is especially true for DLSw networks.

General Broadcast Frame Filtering

NetBIOS stations rarely use General Broadcast frames. However, when they do, it's to send data frames that can be filtered to all NetBIOS stations on a network. The NetBIOS General Broadcast frame is Datagram-Broadcast.

Trace Control Frames Frame Filtering

Trace Control frames rarely use NetBIOS traces, but when they do they terminate said traces in all NetBIOS stations on a network. These can typically be filtered. The NetBIOS Trace Control frame is Terminate-Trace.

11.2.2 Configuring frame type filtering

For bridge traffic, the router does not filter any of the above frame by default. However, if you are bridging NetBIOS traffic on WAN links, it may be helpful to filter these frames. To turn frame filtering on or off for bridging, run **set filters bridge**.

For DLSw traffic, the router filters all the above frame types by default. To turn frame type filtering on or off for DLSw, run **set filters dlsw**.

Syntax:

```
NetBIOS config>set filters bridge <flt_nm_cnflct_frms> <flt_gnrl_brdcst_frms> <flt_trc_cntrl_frms>
```

Example:

Activating name conflict resolution frame filtering, deactivating the general broadcast frame filtering and finally activating the trace control frames for bridge traffic.

```
NetBIOS config>set filters bridge yes no yes

NetBIOS config>
```

11.2.3 Duplicate frame filtering

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

Duplicate frame filtering causes the router to forward only one instance for each frame within a configurable amount of time. *Fig. 21* on page 100 shows how duplicate frame filtering reduces the number of broadcast frames forwarded over the DLSw WAN.

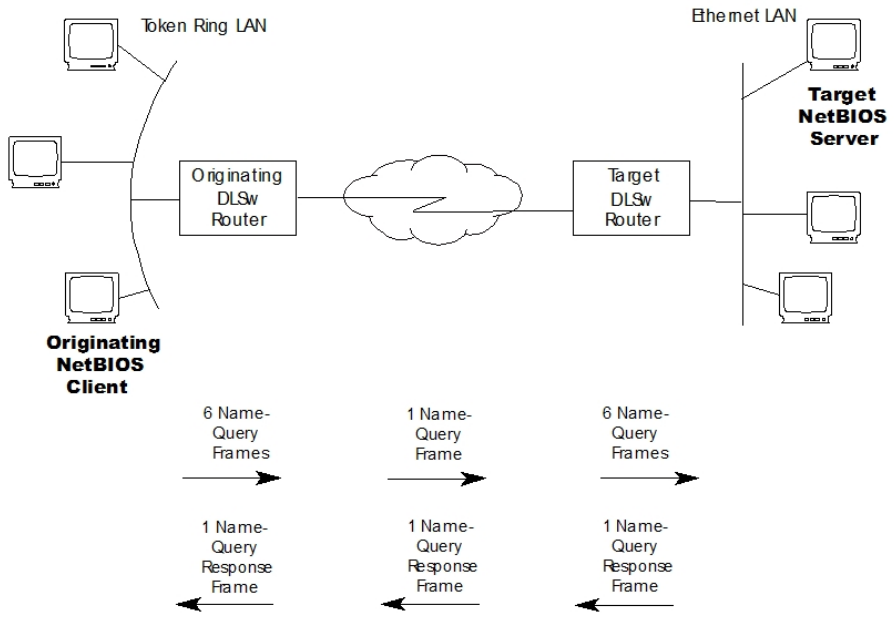

Fig. 21: **Setting Up a NetBIOS Session over DLSw**

The following is the process the originating NetBIOS client uses to set up a session with the target NetBIOS server.

(1) After verifying the 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 if it were the originating NetBIOS station.

(4) The target NetBIOS station responds to the Name-Query frames with a corresponding Name-Recognized frame containing 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.

11.2.4 How duplicate frame filtering works

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

Fig. 22 on page 101 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.

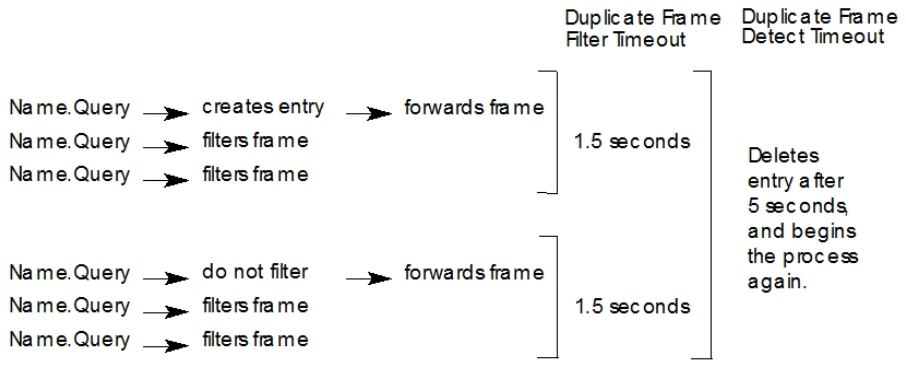

Fig. 22: **Duplicate Frame Filtering Process for Bridged Traffic**

Here are the steps for duplicate frame filtering:

(1) When the router receives a new frame, it creates an entry for said frame in the duplicate frame database and forwards it.

(2) The router filters any duplicate frames 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 times out, it forwards the frame and resets the timer.

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

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 times out. For DLSw, you can also control how many query frames the target DLSw router sends during a configurable time period.

11.2.5 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 by default. You can enable or disable it by running **enable duplicate-filtering** and **disable duplicate-filtering**.

To change the timers, run **set general**.

Syntax:

```
NetBIOS config>set general <dup_frm_flt_tmt> <dup_frm_flt_tmt> <cmd_frm_rtry_cnt>
                           <cmd_frm_rtry_tmt>
  <0..1000>   Duplicate frame filter timeout (1/10 secs.)
  <10..1000>  Duplicate frame detect timeout (1/10 secs.)
  <0s..10s>   Command frame retry count
  <0..100>    Command frame retry timeout (1/10 secs.)
```

Example:

```
NetBIOS config>set general 15 50 0 0

NetBIOS config>
```

Warning

Setting Duplicate Frame Filter Timeout to zero…

Disables duplicate frame checking!

For DLSw, the command frame retry count [5] and command frame retry timeout value in seconds [0.5] allows you to control how many query frames are sent by the destination DLSw router during a configured period of time.

11.2.6 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 for-

ward it.

You cannot disable response frame filtering on the router.

11.2.7 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 *Duplicate frame filtering* on page 100, 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 said 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 by running **set general**.

Syntax:

```
NetBIOS config>set general <dup_frm_flt_tmt> <dup_frm_flt_tmt> <cmd_frm_rtry_cnt>
                           <cmd_frm_rtry_tmt>
  <0..1000>   Duplicate frame filter timeout (1/10 secs.)
  <10..1000>  Duplicate frame detect timeout (1/10 secs.)
  <0s..10s>   Command frame retry count
  <0..100>    Command frame retry timeout (1/10 secs.)
```

Warning

Setting Duplicate Frame Filter Timeout to zero…

Disables duplicate frame checking!

11.2.8 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 Status-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 times out.

11.2.9 Enabling caching

Name caching is always enabled, you cannot disable it. The default for route caching is disabled. Run **enable route-caching** to enable it.

```
NetBIOS config>enable route-caching

Route caching is                        ON

NetBIOS config>
```

11.2.10 Types of name cache entries

There are three types of name cache entries:

- *Permanent* entries are those you add in the NetBIOS configuration menu. The router saves permanent entries and they are still available when you restart the router.
- *Static* entries are those you enter in the NetBIOS monitoring menu. The router does not save static entries and they are not available after you restart the router.

- *Dynamic* entries are those the router learns through Name-Query and Name-Recognized processing. A timer removes dynamic entries that are not referenced within a configurable period 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 on the name, indicating 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 said 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.

11.2.11 Adding name cache entries

Adds permanent or static entries for DLSw neighbors to name caching. Although the router lets you add entries other than DLSw neighbors, it ignores said 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 (>).

Entering **add cache-entry** in the NetBIOS menu to add static entries.

Syntax:

```
NetBIOS config>add cache-entry <nbname> <ip>
  <word>      Enter up to 15 characters of NetBIOS name
  <a.b.c.d>   Ipv4 format
```

Example:

Creation of an entry associated with name *nbs*, with IP address 172.24.52.23.

```
NetBIOS config>add cache-entry nbs 172.24.52.23

Name cache entry has been created

NetBIOS config>
```

11.2.12 Setting cache parameters

Use **set cache-parms** to change the different cache operating parameters.

Syntax:

```
NetBIOS config>SET CACHE-PARMS <sgnfcnt_chrs> <bst_pth> <rdc_srch_tmt>
<uref_entry_tmt> <max_loc> <max_rem>
  [15, 16]      Significant characters in name [16]
  <10..1000000> Best path aging timeout value (1/10 secs.) [60.0]
  <10..1000>    Reduced search timeout value (1/10 secs.) [1.5]
  <1..100000>   Unreferenced entry timeout value in minutes [5000]
  <100..30000>  Max nbr local name cache entries [500]
  <100..30000>  Max nbr remote name cache entries[100]
```

See *NetBIOS Filtering and Caching commands* on page 105, *SET* on page 114 command for more information on the **set cache-parms** command.

11.2.13 Displaying cache entries

From the NetBIOS configuration menu, run the **list cache** commands found in *NetBIOS List Cache Configuration Commands* on page 103 to view the cache contents.

NetBIOS List Cache Configuration Commands

Command	Displays

LIST CACHE ALL	All active entries in the router 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 menu, run the **list cache** commands found in *NetBIOS List Cache Monitoring Commands* on page 104 to view the cache contents.

NetBIOS List Cache Monitoring Commands

Command	Displays
LIST CACHE ACTIVE	All active entries in the router 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 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 the router learns over the DLSw WAN.
LIST CACHE UNKNOWN	Entries where the types of NetBIOS entry is unknown.

11.2.14 NetBIOS name filtering

NetBIOS name filters apply to both bridging and DLSw. Use them to filter NetBIOS packets with specific NetBIOS host names. The router examines the source name or destination name field for 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 *Configuration and Monitoring NetBIOS Name and Byte Filters* on page 130.

11.2.15 NetBIOS byte filtering

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

To build a byte filter, 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 *Configuration and Monitoring NetBIOS Name and Byte Filters* on page 130.

Chapter 12 NetBIOS Filtering and Caching commands

12.1 About NetBIOS configuration and monitoring commands

Any changes made in the configuration menu do not take immediate effect: restart the router to active them.

Monitoring commands take effect immediately, however, the router does not save them after you restart the router.

12.2 Configuring NetBIOS filtering and caching

Configures the following NetBIOS filtering and caching parameters:

- To configure name caching parameters, enter **set cache-parms**.
- To configure duplicate frame filtering, enter **set general**.
- To configure frame type filtering, enter **set filters bridge** or **set filters dlsw**.

12.2.1 Configuring NetBIOS for DLSw

If you are sending NetBIOS traffic over DLSw, you can also:

- 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.

12.2.2 Adding name cache entries for DLSw neighbors

Adds 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 *ADD* on page 108 command for more information. NetBIOS names are case sensitive and must match the network NetBIOS names.

Example:

```
NetBIOS config>add cache-entry accounting<0000> 135.77.25.2

Name cache entry has been created

NetBIOS config>add cache-entry <686f7374> 10.20.30.40

Name cache entry has been created

NetBIOS config>list cache all

Entry  Name                 IP Address
-----  -------------------  ---------------
    1  accounting<0000>   <03>
                            10.2.1.2
    2  host           <03>  10.20.30.40

NetBIOS config>
```

12.2.3 Opening NetBIOS SAPs

Run **open-sap** (DLSw configuration menu), to open NetBIOS SAPs on both sides of the link, to enable DLSw to transmit NetBIOS frames.

Syntax:

```
DLSw config>open-sap <interface>  <SAP in hex (range 0-F4) | sna | nb | lnm>
```

```
  <hex 0..f4>    SAP number
  sna            Open SNA SAPs
  nb             Open NB SAP
  lnm            Open LNM SAP
DLSw config>
```

Example:

```
DLSw config>open-sap ethernet0/0 4
DLSw config>
```

12.2.4 Setting a priority for SNA and NetBIOS sessions

Prioritizes SNA and NetBIOS traffic to prevent one type of session from using too much of the available bandwidth during network congestion.

To do this, set the SNA traffic priority, NetBIOS traffic priority and the priority queues management by running **sna-priority**, **nbs-priority** and **dls-queses**.

Syntax:

```
DLSw config>sna-priority ?
  critical
  high
  low
  medium
```

```
DLSw config>nbs-priority ?
  critical
  high
  low
  medium
```

```
DLSw config>dls-queues <priority_class> <msg_allocation>
<priority_class>
  critical    Configure critical queue priority
  high        Configure high queue priority
  medium      Configure medium queue priority
  low         Configure low queue priority
<msg_allocation>
  <1..9>    Value in the specified range
```

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.

By default, the message number assignation for each priority is 4/3/2/1 (4 messages for Critical priority for each one of Low priority). The number of messages assigned to the Critical, High, Medium and Low priorities (values from 1 to 9), must be configured in descending order: the higher the priority, the higher the number of messages processed.

12.2.5 Setting the maximum NetBIOS frame size

To change the maximum NetBIOS frame size, run **nbs-mtu-ui-frames** from the DLSw configuration menu. Default is 2052. Set this parameter to the largest frame size you expect to need and no larger. Setting the frame size larger than required, reduces the number of available buffers.

Syntax:

```
DLSw config>nbs-mtu-ui-frames ?
  516
  1470
  2052
  4399
```

12.2.6 Setting the memory allocation for NetBIOS UI frames

Enter **nbs-global-memory** in the DLSw configuration menu prompt to set the number of bytes the router allocates as a buffer for NetBIOS UI frames. If the TCP transmit buffer is full, the router uses the former buffer to collect NetBIOS UI frames.

Note: the number of bytes allocated for NetBIOS is global, not per session.

Syntax:

```
DLSw config>nbs-global-memory ?
  <0..4294967295>    Netbios UI-Frames memory space
```

12.3 Configuring NetBIOS

12.3.1 Accessing the NetBIOS configuration menu

Accesses the NetBIOS configuration menu from the main bridge instance configuration menu, from the configuration menu of any of the virtual bridge entities or from the DLSw configuration menu.

The configuration menu is common to both the DLSw and the main bridge instance. Configuration changes executed in either of the two menus affect both the DLSw as well as the NetBIOS operations over the main bridge instance. However, changes made in a virtual bridge entity configuration menu only affect said entity.

(1) To access the NetBIOS configuration menu for a bridge entity, run **netbios** from the configuration menu for said entity.

Example 1:
Accessing the main entity configuration menu.

```
Config>protocol asrt

-- ASRT Bridge user configuration --
ASRT config>netbios

-- NetBIOS Support User Configuration --
NetBIOS config>
```

Example 2:
Accessing the virtual bridge entity configuration menu with identifier 2.

```
Config>protocol asrt

-- ASRT Bridge user configuration --
ASRT config>virtual-bridge 2

-- Virtual ASRT Bridge user configuration --
VBDG config>netbios

-- NetBIOS Support User Configuration --
VBDG NetBIOS config>
```

(2) To access the NetBIOS configuration menu for the DLSw protocol, run **netbios** from the configuration for said protocol.

```
Config>protocol dls

-- DLSw protocol user configuration --
DLSw config>netbios

-- NetBIOS Support User Configuration --
NetBIOS config>
```

12.3.2 NetBIOS configuration commands

The commands available in the NetBIOS configuration menu are detailed below:

Command	Function
? (HELP)	Displays the configuration commands or their options.
ADD	Adds entries to the device name cache.

DELETE	Deletes entries from the device name cache.
DISABLE	Deactivates duplicate frame filtering or route caching.
ENABLE	Activates duplicate frame filtering or route caching.
LIST	Displays configuration information.
SET	Configures different parameters associated with NetBIOS operations.
EXIT	Exits the NetBIOS configuration menu.

12.3.3 ? (HELP)

Lists the available commands or their options.

Syntax:

```
NetBIOS config>?
```

Example:

```
NetBIOS config>?
  add        Add a new cache entry for DLSw neighbors
  delete     Delete a cache entry
  disable    Disable netbios features
  enable     Enable netbios features
  list       List configuration
  set        Set NetBIOS parameters
  exit
NetBIOS config>
```

12.3.4 ADD

Adds a new entry to the device name cache.

Syntax:

```
NetBIOS config>add ?
  cache-entry    Add a new cache entry for DLSw neighbors
```

12.3.4.1 ADD CACHE-ENTRY

Adds a new entry to the router name cache. You can only add name cache entries for DLSw neighbors. The router ignores entries for bridge 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 network NetBIOS names.

Syntax:

```
NetBIOS config>add cache-entry <name> <ip-address>
```

Example:

```
NetBIOS config>add cache-entry accounting<0000> 135.77.25.2

Name cache entry has been created

NetBIOS config>list cache all

Entry  Name                 IP Address
-----  -------------------  ---------------
    1  nbs            <00>  172.24.52.23
    2  accounting<0000>   <03>
```

```
                          10.2.1.2

NetBIOS config>
```

12.3.5 DELETE

Deletes an entry from the device name cache entries. Specify the entry number you want to delete. To see a list of entry numbers, run **list cache all**.

Syntax:

```
NetBIOS config>delete cache-entry <Cache record number:(1..65535)>
```

Example:

```
NetBIOS config>list cache all

Entry  Name                 IP Address
-----  -------------------  ---------------
    1  host           <03>  10.20.30.40
    2  accounting<0000>   <03>
                          10.2.1.2
    3  nbs            <03>  172.24.52.23

NetBIOS config>delete cache-entry 2

Name cache entry has been deleted

NetBIOS config>list cache all

Entry  Name                 IP Address
-----  -------------------  ---------------
    1  host           <03>  10.20.30.40
    2  nbs            <03>  172.24.52.23

NetBIOS config>
```

12.3.6 DISABLE

Disables duplicate frame filtering or route caching for the bridge.

Syntax:

```
NetBIOS config>disable ?
  duplicate-filtering    Disable duplicate frame filtering for bridging
  route-caching          Disable route caching for bridging
NetBIOS config>
```

12.3.6.1 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>
```

12.3.6.2 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>
```

12.3.7 ENABLE

Enables duplicate frame filtering or route caching for the bridge.

Syntax:

```
NetBIOS config>enable ?
  duplicate-filtering    Enable duplicate frame filtering for bridging
  route-caching          Enable route caching for bridging
NetBIOS config>
```

12.3.7.1 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>
```

12.3.7.2 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>
```

12.3.8 LIST

Displays the configuration information.

Syntax:

```
NetBIOS config>list ?
  cache      List cache entries
  filters    List bridging and DLSw filtering state
  general    List current NetBIOS caching and filtering configuration
NetBIOS config>
```

12.3.8.1 LIST CACHE

Displays information on the name cache.

Syntax:

```
NetBIOS config>list cache ?
  all             List all cache entries
  entry-number    List cache entries by entry number
  ip-address      List cache entries by ip
  name            List cache entries by name
NetBIOS config>
```

12.3.8.1.1 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, shown just before the IP address, is the 16th character of the NetBIOS name. IBM and Microsoft reserve said 16th character and it always appears in hexadecimal.

Example:

```
NetBIOS config>list cache all

Entry  Name                IP Address
-----  ------------------  ---------------
    1  test          <00>  1.2.3.4
    2  example       <00>  145.67.89.10

NetBIOS config>
```

12.3.8.1.2 LIST CACHE ENTRY-NUMBER

Displays a cache entry according to its entry number. Run **list cache all** to see a list of all entry numbers.

Example:

```
NetBIOS config>list cache entry-number 2

Entry  Name                IP Address
-----  ------------------  ---------------
    2  example       <00>  145.67.89.10

NetBIOS config>
```

12.3.8.1.3 LIST CACHE IP-ADDRESS

Displays an entry for a specific IP address.

Example:

```
NetBIOS config>list cache ip-address 145.67.89.10

Entry  Name                IP Address
-----  ------------------  ---------------
    2  example       <00>  145.67.89.10

NetBIOS config>
```

12.3.8.1.4 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 for wildcards matching San Francisco:

Fran	S??*?????????
San?Fran?isco	S?*
*S**	S?n?F?a?c?s?o?
**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, depend-

ing on how many significant characters were configured through **set cache-parms**).

Note

To avoid problems when entering the “?” wildcard, use quotation marks around the name you wish to search for.

When using the “$” wildcard, include the final spaces in the name.

NetBIOS are case sensitive.

Example:

```
NetBIOS config>list cache all

Entry  Name                IP Address
-----  ------------------  ---------------
    1  host          <aa>  1.2.3.4
    2  hsst          <aa>  1.2.3.4
    3  San Francisco <aa>  2.3.4.5
    4  San2Fr        <03>  14.32.12.2

NetBIOS config>list cache name "San?F*"

Entry  Name                IP Address
-----  ------------------  ---------------
    3  San Francisco <aa>  2.3.4.5
    4  San2Fr        <03>  14.32.12.2

NetBIOS config>list cache name s*

Name cache entry NOT found for name entered

NetBIOS config>list cache name "isco$"

Name cache entry NOT found for name entered

NetBIOS config>list cache name "isco  $"

Entry  Name                IP Address
-----  ------------------  ---------------
    3  San Francisco <aa>  2.3.4.5

NetBIOS config>
```

12.3.8.2 LIST FILTERS

Lists the status of the all configured filters.

Syntax:

```
NetBIOS config>list filters ?
  all       List bridging and DLSw filtering state
  bridge    List bridging filtering state
  dlsw      List DLSw filtering state
NetBIOS config>
```

12.3.8.2.1 LIST FILTERS ALL

Indicates whether frame type filtering is on or off for both bridging and DLSw. Use **set filters bridge** and **set filters dlsw** 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>
```

12.3.8.2.2 LIST FILTERS BRIDGE

Indicates whether frame type filtering is on or off for bridging. Enter **set filters bridge** 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>
```

12.3.8.2.3 LIST FILTERS DLSW

Indicates whether frame type filtering is on or off for 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>
```

12.3.8.3 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.

12.3.9 SET

Configures the different parameters associated with the NetBIOS functionality.

Syntax:

```
NetBIOS config>set ?
  cache-parms    Configure cache parameters
  filters        Configure filter parameters
  general        Configure general parameters
NetBIOS config>
```

12.3.9.1 SET CACHE-PARMS

Sets name caching parameters that apply to bridging or DLSw.

Syntax:

```
NetBIOS config>SET CACHE-PARMS <sgnfcnt_chrs> <bst_pth> <rdc_srch_tmt> <uref_entry_tmt> <max_loc> <max_rem>
  sgnfcnt_chrs     [15, 16]      Significant characters in name [16]
  bst_pth          <10..1000000> Best path aging timeout value (1/10 secs.) [60.0]
  rdc_srch_tmt     <10..1000>    Reduced search timeout value (1/10 secs.) [1.5]
  uref_entry_tmt   <1..100000>   Unreferenced entry timeout value in minutes [5000]
  max_loc          <100..30000>  Max nbr local name cache entries [500]
  max_rem          <100..30000>  Max nbr remote name cache entries[100]
```

Example:

```
NetBIOS config>set cache-parms 16 50 20 6000 400 200
```

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. 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 times out, 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. 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 times out, it presumes the previous route is no longer valid and 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. 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 said length of time before deleting it. If the cache fills up, the router removes entries sooner. 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 the router learns over the bridge. 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) 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. 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%.

12.3.9.2 SET FILTERS

Syntax:

```
NetBIOS config> set filters ?
  bridge       Configure frame-type filtering for bridging
  byte-name    Display the NetBIOS filtering prompt
  dlsw         Configure frame-type filtering for DLSw traffic
NetBIOS config>
```

12.3.9.2.1 SET FILTERS BRIDGE

Configures the frame-type filtering for bridging.

Syntax:

```
NetBIOS config>SET FILTERS BRIDGE <flt_nm_cnflct_frms> <flt_gnrl_brdcst_frms> <flt_trc_cntrl_frms>
```

flt_nm_cnflct_frms	Name conflict resolution frame filtering.
flt_gnrl_brdcst_frms	General broadcast frame filtering.
flt_trc_cntrl_frms	Trace control frame filtering.

Example:

Activates the name conflict resolution frame filtering, deactivates the general broadcast frame filtering and finally, activates the trace control frame filtering for bridge traffic.

```
NetBIOS config>set filter bridge yes no yes
NetBIOS config>
```

12.3.9.2.2 SET FILTERS BYTE-NAME

Accesses the NetBIOS frame name and byte filtering configuration menu.

See *Configuration and Monitoring NetBIOS Name and Byte Filters* on page 130 for more information on the commands available in this menu.

Example:

```
NetBIOS config>set filters byte-name

-- NETBIOS Filtering configuration --
NETBIOS Filter config>
```

12.3.9.2.3 SET FILTERS DLSW

Configures the frame-type filters for DLSw traffic.

Syntax:

```
NetBIOS config>SET FILTERS DLSw <flt_nm_cnflct_frms> <flt_gnrl_brdcst_frms> <flt_trc_cntrl_frms>
```

flt_nm_cnflct_frms	Name conflict resolution frame filtering.
flt_gnrl_brdcst_frms	General broadcast frame filtering.
flt_trc_cntrl_frms	Trace control frame filtering.

Example:

Activates the name conflict resolution frame filtering, deactivates the general broadcast frame filtering and finally, activates the trace control frame filtering for DLSw traffic.

```
NetBIOS config>set filters dlsw yes no yes
```

```
NetBIOS config>
```

12.3.9.3 SET GENERAL

Configures the duplicated frame filtering operation parameters. Check *Using NetBIOS* on page 98 *Duplicate frame filtering* on page 100 for more information on how duplicate frame filters work.

Syntax:

```
NetBIOS config>set general <dup_frm_flt_tmt> <dup_frm_flt_tmt> <cmd_frm_rtry_cnt> <cmd_frm_rtry_tmt>
  <0..1000>   Duplicate frame filter timeout (1/10 secs.)
  <10..1000>  Duplicate frame detect timeout (1/10 secs.)
  <0s..10s>   Command frame retry count
  <0..100>    Command frame retry timeout (1/10 secs.)
```

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. Default is 1.5 seconds.
Duplicate frame detect timeout	Timeout time to detect duplicate frames. Applies to both bridged and DLSw traffic. Amount of time the router saves entries in its duplicate frame filter database. When this times out, the router creates new entries for new frames that it receives. The range is 0.0 to 100.000 seconds. 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. Default is 5 seconds.
Command frame retry timeout	Applies to DLSw traffic. 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. Default is 5 seconds.

Example:

```
NetBIOS config>set general 14 50 6 5
NetBIOS config>
```

Warning

Setting Duplicate Frame Filter Timeout to zero…

disables duplicate frame checking!

12.3.10 EXIT

Exits the NetBIOS configuration menu.

Syntax:

```
NetBIOS config>exit
```

Example:

```
NetBIOS config>exit
ASRT config>
```

12.4 NetBIOS monitoring

12.4.1 Accessing the NetBIOS monitoring menu

You can access the NetBIOS monitoring menu from the main bridge entity monitoring menu, from the monitoring menu of any of the virtual bridge entities or from the DLSw monitoring menu.

The monitoring menu is common to both the DLSw and the main bridge entity. Monitoring changes executed in either of the two menus affect both the DLSw as well as the NetBIOS operations over the main bridge entity; otherwise, changes made in a virtual bridge entity monitoring menu only affect said entity.

(1) To access the NetBIOS monitoring menu for a bridge entity, run **netbios** from the monitoring menu of said entity.

Example 1:
Accessing the main entity monitoring menu.

```
+protocol asrt

ASRT+virtual-bridge 0

ASRT Main Bridge+netbios

NetBIOS Support User Console

NetBIOS+
```

Example 2:
Accessing the virtual bridge entity monitoring menu with identifier 2.

```
+protocol asrt

ASRT+virtual-bridge 1

ASRT Virtual Bridge 1+netbios

NetBIOS Support User Console

NetBIOS+
```

(2) To access the NetBIOS monitoring menu for the DLSw protocol, run **netbios** from the monitoring menu for said protocol.

```
+protocol dls
Data Link Switching Console
DLSw+netbios

NetBIOS Support User Console

NetBIOS+
```

12.4.2 NetBIOS monitoring commands

The commands available in the NetBIOS monitoring menu are detailed below:

Command	Function
? (HELP)	Displays the monitoring commands or their options.
ADD	Adds entries to the device name cache.
DELETE	Deletes entries from the device name cache.
DISABLE	Deactivates duplicate frame filtering or route caching.
ENABLE	Activates duplicate frame filtering or route caching.
LIST	Displays the NetBIOS operating information.
SET	Configures different parameters associated with NetBIOS operations.
EXIT	Exits the NetBIOS monitoring menu.

12.4.3 ? (HELP)

Lists available commands or options.

Syntax:

```
NetBIOS+?
```

Example:

```
NetBIOS+?
  add         Adds a new entry
  delete      Deletes an entry
  disable     Disables duplicate frame filtering or route caching for the bridge
  enable      Enables duplicate frame filtering or route caching for the bridge
  list        Lists information about NETBIOS operation
  set         Sets different parameters related to the operation of NetBIOS
  exit
```

12.4.4 ADD

Adds a new name cache entry to the router static configuration.

Syntax:

```
NetBIOS+add ?
cache-entry    Add a new cache entry for DLSw neighbours
```

12.4.4.1 ADD CACHE-ENTRY

Adds a new entry to the router name cache. You can add name cache entries for DLSw neighbors only. The router ignores entries for bridge 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 network NetBIOS names.

Syntax:

```
NetBIOS+add cache-entry <name> <ip-address>
```

Example:

```
NetBIOS+add cache-entry accounting<0000> 135.77.25.2

Name cache entry has been created

NetBIOS+
```

12.4.5 DELETE

Deletes name cache entries from the router static configuration or active cache. You need to specify the name associated with the cache entry you wish to delete. To see a list of entries, run **list cache conf** or **list cache active**.

Note

NetBIOS names are case sensitive.

Syntax:

```
NetBIOS+delete ?
  cache-entry    Deletes NetBIOS name cache entries
NetBIOS+delete cache-entry ?
  <word>    NetBIOS name for cache entry
```

Example:

```
NetBIOS+delete cache-entry accounting<0000>
```

```
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+
```

12.4.6 DISABLE

Disables duplicate frame filtering or route caching for the bridge.

Syntax:

```
NetBIOS+disable ?
  duplicate-filtering    Disables NetBIOS duplicate frame filtering
  route-caching          Disables NetBIOS route caching
```

12.4.6.1 DISABLE DUPLICATE-FILTERING

Disables duplicate frame filtering for bridging. Duplicate frame filtering is always enabled for DLSw traffic (it cannot be enabled or disabled).

Example:

```
NetBIOS+disable duplicate-filtering

Duplicate frame filtering is           OFF

NetBIOS+
```

12.4.6.2 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 (it cannot be enabled or disabled).

Example:

```
NetBIOS+disable route-caching
Route caching is                       OFF
NetBIOS+
```

12.4.7 ENABLE

Enables duplicate frame filtering or route caching for the bridge.

Syntax:

```
NetBIOS+enable ?
  duplicate-filtering    Enables NetBIOS duplicate frame filtering
  route-caching          Enables NetBIOS route caching
```

12.4.7.1 ENABLE DUPLICATE-FILTERING

Enables duplicate frame filtering for bridging. Duplicate frame filtering is always enabled for DLSw traffic (it cannot be enabled or disabled).

Example:

```
NetBIOS+enable duplicate-filtering
Duplicate frame filtering is            ON
NetBIOS+
```

12.4.7.2 ENABLE ROUTE-CACHING

Enables route caching for bridging. Route caching is always enabled for DLSw traffic (it cannot be enabled or disabled). 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+
```

12.4.8 LIST

Displays information on the NetBIOS operations.

Syntax:

```
NetBIOS+list ?
  cache         Lists information about the cache names
  filters       Lists the state of the configured filters
  general       Lists NetBIOS general configuration information
  statistics    Lists NetBIOS statistics
```

12.4.8.1 LIST CACHE

Displays information on the name cache.

Syntax:

```
NetBIOS+list cache ?
  active     Lists all NetBIOS name cache information
  config     Lists all statics and permanents entries from the cache names
  group      Lists NetBIOS name cache information for name groups
  local      Lists NetBIOS name cache information for local names
  name       Lists NetBIOS name cache detail information
  remote     Lists NetBIOS name cache information for remote names
  unknown    Lists NetBIOS name cache information for unknown names
```

12.4.8.1.1 LIST CACHE ACTIVE

Displays all active entries in the router name cache, including dynamic, static and permanent entries.

The router displays all hexadecimal data in angle brackets. The number in angle brackets, shown just before the IP address, is the 16th character of the NetBIOS name. IBM and Microsoft reserve the 16th character 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 in 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+
```

12.4.8.1.2 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, shown just before the IP address, is the 16th character of the NetBIOS name. IBM and Microsoft reserve the 16th character and it always appears in hexadecimal.

Example:

```
NetBIOS+list cache config
```

```
Cnt  NetBIOS Name         Entry Type  Rem Path St  IP Address(es)
---  -------------------  ----------  -----------  ---------------
  1  ID             <1d>  DYNAMIC     GROUP

NetBIOS+
```

12.4.8.1.3 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 in this section.

Example:

```
NetBIOS+list cache group

Cnt  NetBIOS Name         Entry Type  Rem Path St  IP Address(es)
---  -------------------  ----------  -----------  ---------------
  1  ID             <1d>  DYNAMIC     GROUP

NetBIOS+
```

12.4.8.1.4 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 in 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+
```

12.4.8.1.5 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 matching San Francisco:

Fran	S??*?????????
San?Fran?isco	S?*
S*	S?n?F?a?c?s?o?
*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

To enter the “?” wildcard without difficulty, use quotation marks around the name you wish to search for.

When using the “$” wildcard, include the final spaces in the name.

NetBIOS are case sensitive.

Syntax:

```
NetBIOS+list cache name ?
  <word>    NetBIOS name for cache entry
```

Example:

```
NetBIOS+list cache name 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 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	Permanent entries created in the configuration process.
	STATIC	Permanent entries created in the monitoring process.
	DYNAMIC	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	Number of timer ticks since reference was made to an entry. Timer ticks vary according	
cache last Reference	The number of timer ticks vary according to the hardware platform.	
Local path information:		

Loc Path St	Local Path State. The 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 an entry was last updated. Timer ticks vary according to hardware platform.	
MAC Address	If the entry corresponds to a server, displays the MAC address of the server.	
LSF	Largest Frame Size that the router can use for the entry.	
Routing	Displays standard Routing Information Field (RIF) information.	
InformationRemote PathInformation		
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 the hardware platform.	
LSF	Largest Frame Size that the router can use for the entry.	
IP Address	IP address of the DLSw partner.	

12.4.8.1.6 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 in 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+
```

12.4.8.1.7 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 in 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+
```

12.4.8.2 LIST FILTERS

Displays the status of the configured filters.

Syntax:

```
NetBIOS+list filters ?
  all       Lists the on/off status for both bridge and DLS frame-type
            filtering
  bridge    Lists the on/off status for bridge frame-type filtering
  dlsw      Lists the on/off status for DLSw frame-type filtering
```

12.4.8.2.1 LIST FILTERS ALL

Displays whether or not frame type filtering is on or off for both bridging and DLSw. Run **set filters bridge** and **set filters dlsw** to turn said 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+
```

12.4.8.2.2 LIST FILTERS BRIDGE

Displays whether or not frame type filtering is on or off for bridging. Run **set filters bridge** to turn said 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+
```

12.4.8.2.3 LIST FILTERS DLSW

Displays whether or not frame type filtering is on or off for DLSw. Run **set filters dlsw** to turn said 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+
```

12.4.8.3 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.

12.4.8.4 LIST STATISTICS

Displays NetBIOS statistics.

Syntax:

```
NetBIOS+list statistics ?
  cache      List NetBIOS name cache statistics
  frames     List frames statistics
  general    List general statistics
```

12.4.8.4.1 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+
```

12.4.8.4.2 LIST STATISTICS FRAMES

Syntax:

```
NetBIOS+list statistics frames ?
  bridge    List NetBIOS bridge duplicate frame handling statistics
  dlsw      Lists NetBIOS DLS duplicate frame handling statistics
```

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+
```

12.4.8.4.3 LIST STATISTICS GENERAL

Syntax:

```
NetBIOS+list statistics general ?
  bridge    Lists NetBIOS bridge frame disposition statistics
  dlsw      Lists NetBIOS DLS frame disposition statistics
```

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+
```

12.4.9 SET

Configures different parameters associated with the NetBIOS operations.

Syntax:

```
NetBIOS+set ?
  cache-parms    Sets name caching parameters that apply to bridging or DLSw
  filters        Sets frame-type filtering
  general        Sets NetBIOS duplicate frame handling and retry parameters
```

12.4.9.1 SET CACHE-PARMS

Sets name caching parameters that apply to bridging or DLSw.

Example:

```
NetBIOS+set cache-parms ?
  <15..16>    Number of significant characters in a NetBIOS name
NetBIOS+set cache-parms 15 ?
  <1.0..100000.0>    Best path aging timeout in seconds (only one decimal
                     value)
NetBIOS+set cache-parms 15 60 ?
  <1.0..100.0>    Reduced search timeout in 10ths of seconds (only one decimal
                  value)
```

```
NetBIOS+set cache-parms 15 60 1.5 ?
  <1..100000>    Unreferenced entry timeout in 10ths of seconds
NetBIOS+set cache-parms 15 60 1.5 5000 ?
  <100..30000>    Maximum local name cache entries
NetBIOS+set cache-parms 15 60 1.5 5000 500 ?
  <100..30000>    Maximum remote name cache entries
NetBIOS+set cache-parms 15 60 1.5 5000 500 100
```

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. 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. 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. 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. Default is 5000 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. 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. 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%.

12.4.9.2 SET FILTERS

Syntax:

```
NetBIOS+set filters ?
  bridge       Sets NetBIOS frame-type filtering parameters
  byte-name    Displays NetBIOS Byte or Name filtering parameters
  dlsw         Sets NetBIOS frame-type filtering parameters
```

12.4.9.2.1 SET FILTERS BRIDGE

Configures frame-type filtering for bridging.

Example:

```
NetBIOS+set filters bridge ?
  <0..1>    Filter name conflict frames (0 -> NO, 1 -> YES)
NetBIOS+set filters bridge 0 ?
```

```
  <0..1>    Filter general broadcast frames (0 -> NO, 1 -> YES)
NetBIOS+set filters bridge 0 1 ?
  <0..1>    Filter trace control frames (0 -> NO, 1 -> YES)
NetBIOS+set filters bridge 0 1 0

Name conflict filtering is             OFF

General broadcast filtering is          ON

Trace control filtering is             OFF
NetBIOS+
```

12.4.9.2.2 SET FILTERS BYTE-NAME

Accesses the NetBIOS frame name and byte filtering monitoring menu.

See *Configuration and Monitoring NetBIOS Name and Byte Filters* on page 130 for more information on the commands available in this menu.

Example:

```
NetBIOS+set filters byte-name

NETBIOS Filter+
```

12.4.9.2.3 SET FILTERS DLSW

Sets frame-type filters for DLSw traffic.

Example:

```
NetBIOS+set filters dlsw ?
  <0..1>    Filter name conflict frames (0 -> NO, 1 -> YES)
NetBIOS+set filters dlsw 1 ?
  <0..1>    Filter general broadcast frames (0 -> NO, 1 -> YES)
NetBIOS+set filters dlsw 1 0 ?
  <0..1>    Filter trace control frames (0 -> NO, 1 -> YES)
NetBIOS+set filters dlsw 1 0 0

Name conflict filtering is              ON

General broadcast filtering is         OFF

Trace control filtering is             OFF
```

12.4.9.3 SET GENERAL

Configures the duplicated frame filtering operating parameters. See *Duplicate frame filtering* on page 100 on *Using NetBIOS* on page 98 for more information on how duplicate frame filters work.

Example:

```
NetBIOS+set general ?
  <0.0..100.0>    Duplicate frame filter timeout value in seconds (only one
                  decimal value)
NetBIOS+set general 1.5 ?
  <1.0..100.0>    Duplicate frame detect timeout value in seconds (only one
                  decimal value)
NetBIOS+set general 1.5 5.0 ?
  <0..10>    Command frame retry count
NetBIOS+set general 1.5 5.0 5 ?
  <0.0..10.0>    Command frame retry timeout value in seconds (only one
                 decimal value)
NetBIOS+set general 1.5 5.0 5 0.5
```

Warning

Setting Duplicate Frame Filter Timeout to zero…

disables duplicate frame checking!

If DLSw is **not** enabled, the *retry count* and *retry timeout* values are not asked for:

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. Default is 1.5 seconds.
Duplicate frame detect timeout	Timeout time for detecting duplicate frames. 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. 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. Default is 5 seconds.
Command frame retry timeout	Applies to DLSw traffic. 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. Default is 5 seconds.

12.4.10 EXIT

Exits the NetBIOS monitoring menu.

Syntax:

```
NetBIOS+exit
```

Example:

```
NetBIOS+exit
ASRT+
```

Chapter 13 Configuration and Monitoring NetBIOS Name and Byte Filters

13.1 Accessing the NetBIOS name and byte configuration and monitoring menus

This section describes the NetBIOS Name and Byte filter configuration and monitoring commands.

To access the NetBIOS name and byte filter configuration menu, run **set filters byte-name** (NetBIOS configuration menu).

Example:

```
Config>protocol asrt

-- ASRT Bridge user configuration --
ASRT config>netbios

-- NetBIOS Support User Configuration --
NetBIOS config>set filters byte-name

-- NETBIOS Filtering configuration --
NETBIOS Filter config>
```

To access the NetBIOS name and byte filter monitoring menu, run **set filters byte-name** (NetBIOS monitoring menu).

Example:

```
protocol asrt

ASRT+virtual-bridge 0

ASRT Main Bridge+netbios

NetBIOS Support User Console

NetBIOS+set filters byte-name

NETBIOS Filter+
```

13.2 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 you enter the filter items in.

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: Creating a filter by name:

Use the following procedure as a guideline to create a name filter. Before you begin, access the NetBIOS name and byte filter configuration menu.

```
Config>protocol asrt

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

```
-- NetBIOS Support User Configuration --
NetBIOS config>set filters byte-name

-- NETBIOS Filtering configuration --
NETBIOS Filter config>
```

(1) Create an empty name filter-list.
Run **create name-filter-list** followed by the name you want to give to the filter-list.

```
NETBIOS Filter config>create name-filter-list boston
NETBIOS Filter config>
```

(2) Access the configuration menu for the created filter-list. Run **update** followed by the name of the filter-list.

```
NETBIOS Filter config>update boston

-- Filter List Configuration --
NETBIOS Name boston config>
```

(3) Add filter items to the filter-list.
When you add a filter item, 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 westboro
NETBIOS Name boston config>
```

(4) Verify the filter item entry.
Run **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 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.
If the packet does not match any of the filter items, the router uses the filter-list default condition (*Inclusive* or *Exclusive*). You can change the list default condition by running **default inclusive** or **default exclusive**. For example: so a packet that doesn't match any filtering elements is dropped by default:

```
NETBIOS Name boston config>default exclusive
```

(6) When you finish adding filter items to the filter-list, run **exit** to return to the NetBIOS configuration menu.

```
NETBIOS Name boston config>exit
NETBIOS Filter config>
```

(7) Add the filter-list to your configuration.
Run **filter-on**. When you turn on a name filter, 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) you want included in this filter.
- Optionally, you can add additional filter lists to the filter. Enter **AND** or **OR** followed by a filter-list name. Use the **end** option to stop adding any further filtering lists to the filter.
  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 2 that contain the name *westboro*.

```
NETBIOS Filter config>filter-on input 2 boston end
NETBIOS Filter config>
```

Another example:

```
NETBIOS Filter config>filter-on output 1 boston or newyork end
NETBIOS Filter config>
```

(8) Run **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 in the bridge being configured.
Use the **enable netbios-filtering** command.

```
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, access the NetBIOS name and byte filter configuration menu.

```
Config>protocol asrt

-- ASRT Bridge user configuration --
ASRT config>netbios

-- NetBIOS Support User Configuration --
NetBIOS config>set filters byte-name

-- NETBIOS Filtering configuration --
NETBIOS Filter config>
```

(1) Create an empty byte filter-list.
Use the **create byte-filter-list** command followed by the name to give the filter-list.

```
NETBIOS Filter config>create byte-filter-list westport
NETBIOS Filter config>
```

(2) Access the created filter-list configuration menu. Run **update** followed by the filter-list name.

```
NETBIOS Filter config>update wesport

-- Filter List Configuration --
NETBIOS Byte westport config>
```

(3) Add filter items to the byte filter-list.
When you add a filter item, specify the following parameters in this order:

- *Inclusive* (bridged) or *exclusive* (dropped).
- *Byte offset* is the number of offset bytes (in decimal) in the packet the router is filtering. The offset is counted from the start of the packet's NetBIOS header. Zero specifies that the router examines all bytes in the packet.
- *Hex pattern* is a hexadecimal number the router used as a pattern to compare with the frame 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* is the mask used to compare the pattern with the frame bytes. This parameter (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 hex pattern. If you omit the hex mask, the router considers it to be all binary 1s (i.e., all the frame bytes are considered as is).
The following example adds a filter item to the byte filter-list *westboro*, which 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 0 12345678
NETBIOS Byte westport config>
```

(4) Verify the filter item entry by running **list**.

```
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 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.
If the packet does not match any of the filter items, the router uses the filter-list default condition (*Inclusive* or *Exclusive*). You can change the default condition by running **default inclusive** or **default exclusive**. For example, so a packet not matching any filtering element is dropped by default:

```
NETBIOS Byte westport config>default exclusive
NETBIOS Byte westport config>
```

(6) When you have finished adding filter items to the list, run **exit** to return to the NetBIOS configuration menu.

```
NETBIOS Byte westport config>exit
NETBIOS Filter config>
```

(7) Add the filter to your configuration.
Run the **filter-on** command. When you turn on a byte filter, 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) you want included in this filter.
- Optionally add additional filter-lists to the filter. Enter **AND** and **OR** followed by a filter-list name. Use the **end** option to stop adding any further filter-lists to the filter.
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 3 westport end
NETBIOS Filter config>
```

(8) Verify the new filter.
Run **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 bridge being configured.
Run the **enable netbios-filtering** command.

```
NetBIOS Filter config>enable netbios-filtering
NETBIOS Filter config>
```

13.3 NetBIOS name and byte filter configuration commands

NetBIOS Name and Byte Filter configuration commands on page 134 lists the NetBIOS name and byte filtering configuration commands.

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.
ENABLE	Enables NetBIOS name and byte filtering.
FILTER-ON	Assigns a filter to a specific port to indicate if the filter is applied to NetBIOS packets input or output on the specified port.
LIST	Displays all information concerning created filters.
UPDATE	Accesses the configuration menu for a filter-list, allowing you to add and delete filtering elements from it.
EXIT	Exits the NetBIOS name and byte filter configuration menu.

13.3.1 ? (HELP)

Lists available commands or options.

Syntax:

```
NETBIOS Filter config>?
```

Example:

```
NETBIOS Filter config>?
  create       Create filter lists for NetBIOS filtering
  delete       Delete filters and filter lists
  disable      Disable NetBIOS name and byte filtering
  enable       Enable NetBIOS name and byte filtering
  filter-on    Assign a filter to a specific port
  list         List configuration
  update       Enter the NetBIOS filter-list prompt
  exit
NETBIOS Filter config>
```

13.3.2 CREATE

Creates a byte filter-list or a name filter-list.

Syntax:

```
NETBIOS Filter config>create ?
  byte-filter-list    Create a byte filter list
  name-filter-list    Create a name filter list
NETBIOS Filter config>
```

13.3.2.1 CREATE BYTE-FILTER-LIST

Creates a byte filter-list. Give the list a unique name of up to 16 characters to identify the filter-list.

Syntax:

```
NETBIOS Filter config>create byte-filter-list <Filter List(1..15 char.)>
```

Example:

```
NETBIOS Filter config>create byte-filter-list westport
NETBIOS Filter config>
```

13.3.2.2 CREATE NAME-FILTER-LIST

Creates a name filter-list. Give the list a unique name of up to 16 character to identify the filter-list.

Syntax:

```
NETBIOS Filter config>create name-filter-list <Filter List(1..15 char.)>
```

Example:

```
NETBIOS Filter config>create name-filter-list newyork
NETBIOS Filter config>
```

13.3.3 DELETE

Deletes byte filter-lists, host name filter-lists, and filters.

Syntax:

```
NETBIOS Filter config>delete ?
  byte-filter-list    Delete a NetBIOS byte filter list
  filter              Delete a NetBIOS filter
  name-filter-list    Delete a NetBIOS name filter list
NETBIOS Filter config>
```

13.3.3.1 DELETE FILTER

Deletes a filter.

Syntax:

```
NETBIOS Filter config>delete filter ?
  input     Delete an input NetBIOS filter
  output    Delete an output NetBIOS filter
NETBIOS Filter config>
```

13.3.3.1.1 DELETE FILTER INPUT

Deletes an input filter for a particular port, created with the **filter-on input** command.

Syntax:

```
NETBIOS Filter config>delete filter input <Port Number (1..254)>
```

Example:

```
NETBIOS Filter config>delete filter input 2
NETBIOS Filter config>
```

13.3.3.1.2 DELETE FILTER OUTPUT

Deletes an output filter for a particular port, created with the **filter-on output** command.

Syntax:

```
NETBIOS Filter config>delete filter output <Port Number (1..254)>
```

Example:

```
NETBIOS Filter config>delete filter output 3
NETBIOS Filter config>
```

13.3.3.2 DELETE BYTE-FILTER-LIST

Deletes a byte filter-list.

Syntax:

```
NETBIOS Filter config>delete byte-filter-list <Filter List(1..15 chars)>
```

Example:

```
NETBIOS Filter config>delete byte-filter-list seattle
NETBIOS Filter config>
```

13.3.3.3 DELETE NAME-FILTER-LIST

Deletes a host-name filter-list.

Syntax:

```
NETBIOS Filter config>delete name-filter-list <Filter List(1..15 chars)>
```

Example:

```
NETBIOS Filter config>delete name-filter-list alaska
NETBIOS Filter config>
```

Note

To delete a filter-list, make sure it isn't associated with any configured filter.

13.3.4 DISABLE

Disables NetBIOS name and byte filtering.

Syntax:

```
NETBIOS Filter config>disable ?
  netbios-filtering    Disable NetBIOS name and byte filtering
NETBIOS Filter config>
```

Example:

```
NETBIOS Filter config>disable netbios-filtering
NETBIOS Filter config>
```

13.3.5 ENABLE

Enables NetBIOS name and byte filtering.

Syntax:

```
NETBIOS Filter config>enable ?
  netbios-filtering    Enable NetBIOS name and byte filtering
NETBIOS Filter config>
```

Example:

```
NETBIOS Filter config>enable netbios-filtering
NETBIOS Filter config>
```

13.3.6 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     Assign a filter to incoming packets on a port
  output    Assign a filter to outgoing packets on a port
        <port-number>         Port Number
                <filter list>        Filter list
                        and <filter list>
                        or <filter list>
```

```
                        end
NETBIOS Filter config>
```

13.3.6.1 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 Number is a configured bridging port number on the router. The port number identifies this filter. Indicate additional filter-lists for a port by using the **AND** and **OR** options followed by the name of the filter-list.

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 2 boston and westport end
NETBIOS Filter config>
```

13.3.6.2 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 Number is a configured bridging port number on the router. The port number identifies this filter. Indicate additional filter-lists for a port by using the **AND** and **OR** options followed by the name of the filter-list.

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 1 boston or newyork end
NETBIOS Filter config>
```

13.3.7 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 (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.

13.3.8 UPDATE

Accesses the filter-list configuration menu, allowing you to add or delete filter items from it. For a description of the commands available in this menu, please see *NetBIOS name and byte filter configuration commands* on page 134 and *NetBIOS name and byte filter monitoring commands* on page 138 in this chapter.

Syntax:

```
NETBIOS Filter config>update <filter list>
```

Example:

```
NETBIOS Filter config>update newyork

-- Filter List Configuration --
NETBIOS Name newyork config>
```

13.3.9 EXIT

Exits the NetBIOS name and byte filter configuration menu.

Syntax:

```
NETBIOS Filter config>exit
```

Example:

```
NETBIOS Filter config>exit
NetBIOS config>
```

13.4 NetBIOS name and byte filter monitoring commands

NetBIOS Name and Byte Filter Monitoring commands on page 138 lists the NetBIOS name and byte filtering monitoring commands.

NetBIOS Name and Byte Filter Monitoring commands

Command	Function
? (HELP)	Lists available commands or options.
LIST	Displays all information concerning created filters.
EXIT	Exits the NetBIOS name and byte filter monitoring menu.

13.4.1 ? (HELP)

Lists available commands or options.

Syntax:

```
NETBIOS Filter+?
```

Example:

```
NETBIOS Filter+?
  list    Lists information related to created filters
  exit
NETBIOS Filter+
```

13.4.2 LIST

Displays information on all filters, on bytes, or on name filters.

Syntax:

```
NETBIOS Filter+list ?
  byte-filter-lists    Lists all of the byte filter lists that you have created
  filters              Lists all the filters that you have created and the
                       number of packets the router has filtered
```

```
 name-filter-lists    Lists all of the name filter lists that you have created
```

13.4.2.1 FILTERS LIST BYTE-FILTER-LISTS

Displays the whole of the created byte filter-list.

Example:

```
NETBIOS Filter+list byte-filter-lists

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+
```

13.4.2.2 LIST FILTERS

Lists all the created filters and the number of packets the router filtered as a result of said filters.

```
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+
```

13.4.2.3 LIST NAME-FILTER-LISTS

Displays all the created name filter-lists.

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+
```

13.4.3 EXIT

Exits the NetBIOS name and byte filter monitoring menu.

Syntax:

```
NETBIOS Filter+exit
```

Example:

```
NETBIOS Filter+exit
NETBIOS+
```

13.5 Byte-Filter-List configuration commands

This section describes the commands available in the byte filter-list configuration menu.

Run **update** to access the filter-list configuration menu, followed by the filter-list name (from the NetBIOS name and byte filter configuration menu).

Example:

```
NETBIOS Filter config>create byte-filter-list westport
NETBIOS Filter config>update westport

-- Filter List Configuration --
NETBIOS Byte westport config>
```

Byte filter list configuration commands on page 140 displays the available configuration commands.

Byte filter list configuration commands

Command	Function
? (HELP)	Displays the available configuration commands or their options.
ADD	Adds a filter item to the configured filter-list.
DEFAULT	Establishes the default action for the filter-list.
DELETE	Eliminates a filter item from the configured filter-list.
LIST	Displays the configuration for the filter-list.
MOVE	Reorders filter items within a filter-list.
EXIT	Exits the byte filter-list configuration menu.

13.5.1 ? (HELP)

Displays the available commands or their options.

Syntax:

```
NETBIOS Byte filter-list config>?
  add        Add a filter item to the filter list
  default    Set default filter action
  delete     Delete a filter item from the filter list
  list       List configuration
  move       Move filter items within the filter list
  exit
NETBIOS Byte filter-list config>
```

13.5.2 ADD

Adds a filter item to the configured byte filter-list.

Syntax:

```
NETBIOS Byte filter-list config>add ?
  exclusive    Add an exclusive filter item
  inclusive    Add an inclusive filter item
        <offset>      <0..65535> Byte Offset
               <pattern>      Hex Pattern
                       <cr>
                       <mask>        Hex Mask
NETBIOS Byte filter-list config>
```

exclusive	Exclusive filter item. When the frame coincides with this element, it is dropped.
inclusive	Inclusive filter item. When the frame coincides with this element, it is bridged.
offset	Offset, within the frame, from the start of the NetBIOS header, where comparing with the configured pattern should start to see if the frame coincides with the filter item.

pattern	Hexadecimal pattern (made up of between 2 and 32 characters) used to check if the frame coincides with the filter item.
mask	Hexadecimal mask (made up of between 2 and 32 characters) to apply to the frame before checking with the configured pattern. This parameter is optional. When not included, the device considers that they are all binary 1's (i.e., all the frame bytes are considered as is).

Note

Adding filter items to filter-lists adds to processing time (as it takes time to evaluate each item on the list) and can affect performance in heavy NetBIOS traffic.

The order the filter items are entered 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.

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 said items within the NetBIOS filter-list represent bytes that do not exist in the NetBIOS packet.

The following example shows how to filter Datagram Broadcast Packets.

Example:

```
NETBIOS Byte westport config>add inclusive 4 09
NETBIOS Byte westport config>
```

13.5.3 DEFAULT

Establishes the default action for the filter-list. If no filter items match the contents of the packet the router considers for filtering, the router forwards or drops the packet (depending on the setting).

Syntax:

```
NETBIOS Byte filter-list config>default ?
  exclusive    Set exclude the default filter action
  inclusive    Set include the default filter action
NETBIOS Byte filter-list config>
```

Example 1:

The packets are dropped when the packet contents do not coincide with any filter item.

```
NETBIOS Byte westport config>default exclusive
NETBIOS Byte westport config>
```

Example 2:

The packets are bridged when the packet contents do not coincide with any filter item.

```
NETBIOS Byte westport config>default inclusive
NETBIOS Byte westport config>
```

13.5.4 DELETE

Deletes a filter item from a filter-list. This reorders filter items on the list: run **list** to view the numbers assigned to each filter item.

Syntax:

```
NETBIOS Byte filter-list config>delete <n° filter>
```

Example:

```
NETBIOS Byte westport config>delete 2
NETBIOS Byte westport config>
```

13.5.5 LIST

Displays information relative 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>
```

13.5.6 MOVE

Re-orders filter items within the filter-list. The filter item number specified by *filter-item-number1* , moves and is re-numbered so it is moved to just before *filter-item-number2* . Run **list** to view the number assigned to each filter item.

Syntax:

```
NETBIOS Byte filter-list config>move <filter-item-number1> < filter-item-number2>
```

Example:

```
NETBIOS Byte westport config>list

BYTE Filter List Name: byte
BYTE Filter List Default: Exclusive

 Item #   Inc/Ex   Offset  Pattern              Mask

   1       Inc        4    0x09                 0xff
   2       Ex         2    0x3344               0xffff
   3       Inc        8    0x08                 0xff

NETBIOS Byte westport config>move 1 3
NETBIOS Byte westport config> list

BYTE Filter List Name: byte
BYTE Filter List Default: Exclusive

 Item #   Inc/Ex   Offset  Pattern              Mask

   1       Ex         2    0x3344               0xffff
   2       Inc        8    0x08                 0xff
   3       Inc        4    0x09                 0xff

NETBIOS Byte westport config>
```

13.5.7 EXIT

Exits the byte filter-list configuration menu.

Syntax:

```
NETBIOS Byte filter-list config>exit
```

Example:

```
NETBIOS Byte westport config>exit
NETBIOS Filter config>
```

13.6 Name-Filter-List configuration commands

Lists the commands available in the name filter-list configuration menu.

Access the filter-list configuration menu by running **update** followed by the filter-list identifier in the NetBIOS name and byte filter configuration menu.

Example:

```
NETBIOS Filter config>create name-filter-list boston
NETBIOS Filter config>update boston

-- Filter List Configuration --
NETBIOS Name boston config>
```

Name filter list configuration commands on page 143 displays the available configuration commands.

Name filter list configuration commands

Command	Function
? (HELP)	Displays the available configuration commands or their options.
ADD	Adds a filter item to the configured filter-list.
DEFAULT	Establishes a default action for the filter-list.
DELETE	Eliminates a filter item from the configured filter-list.
LIST	Displays the configuration for the filter-list.
MOVE	Reorders filter items within a filter-list.
EXIT	Exits the name filter-list configuration menu.

13.6.1 ? (HELP)

Displays the available commands or their options.

Syntax:

```
NETBIOS Name filter-list config>?
  add        Add a filter item to the filter list
  default    Set default filter action
  delete     Delete a filter item from the filter list
  list       List configuration
  move       Move filter items within the filter list
  exit
NETBIOS Name filter-list config>
```

13.6.2 ADD

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.

Syntax:

```
NETBIOS Name filter-list config>add ?
  exclusive    Add an exclusive filter item
  inclusive    Add an inclusive filter item
        ascii
               <hostname> <special char>
        hex
               <hostname>
NETBIOS Name filter-list config>
```

exclusive	Exclusive filter item. When the frame coincides with this, it is dropped.
inclusive	Inclusive filter item. When the frame coincides with this, it is bridged.
ascii	Select this option to enter the NetBIOS name as a chain of 16 ASCII characters.

	This can contain any character except the following: **. / \ [] : | < > + = ; ,** space. Use **?** to indicate a single character wildcard. Use * as the final character in the name to indicate a wildcard for the rest of the name. If the name has less than 15 characters, it is padded to the fifteenth character with ASCII spaces.
hex	Select this option to enter the NetBIOS name as a chain of 32 ASCII characters (16 hexadecimal numbers). The name must have an even number of characters. Use the **??** to specify a wildcard for a single byte. If 32 characters are not introduced, then it is padded to the fifteenth byte with ASCII spaces and with a wildcard for the sixteenth byte (numbers 31º and 32º).
hostname	NetBIOS name to use for name filter.
special char	Optional parameter to indicate the NetBIOS name sixteenth character. This can be used if the name has less than 16 characters. This is a hexadecimal character indicating the value for the last character. If this is not specified in a name of less than sixteen characters, the device uses a **?** wildcard for the sixteenth character.

Example 1:

```
NETBIOS Name boston config>add inclusive ascii qwerty
NETBIOS Name boston config>
```

Example 2:

```
NETBIOS Name boston config>add exclusive hex abc123987fed
NETBIOS Name boston config>
```

13.6.3 DEFAULT

Establishes the default action for the filter-list. 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 ?
  exclusive    Set include the default filter action
  inclusive    Set exclude the default filter action
NETBIOS Name filter-list config>
```

Example:

The packets are bridged when the contents do not coincide with any filtering element.

```
NETBIOS Name boston config>default inclusive
NETBIOS Name boston config>
```

13.6.4 DELETE

Deletes a filter item from the list. This reorders the filter items on a list. Run **list** to check the number assigned to each filter item.

Syntax:

```
NETBIOS Name filter-list config>delete <n° filter>
```

Example:

```
NETBIOS Name boston config>delete 4
NETBIOS Name boston config>
```

13.6.5 LIST

Displays information relative to items in a 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>
```

13.6.6 MOVE

Re-orders filter items within the filter-list. The filter item specified by *filter-item-number1* , moves and is renumbered so it is moved to just before *filter-item-number2* . Run **list** to view numbers assigned to each filter item.

Syntax:

```
NETBIOS Name filter-list config>move <filter-item-number1> <filter-item-number2>
```

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 1 3
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      seattle
   2      HEX      Ex       abc123987fed
   3      ASCII    Inc      westboro

NETBIOS Name boston config>
```

13.6.7 EXIT

Exits the name filter-list configuration menu.

Syntax:

```
NETBIOS Name filter-list config>exit
```

Example:

```
NETBIOS Name boston config>exit
NETBIOS Filter config>
```

Chapter 14 Using MAC Filtering

14.1 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 processed, filtered, or tagged:

- *Processed* - Packets pass through the bridge unaffected.
- *Filtered* - Packets *do not* pass through the bridge (they're dropped).
- *Tagged* - Packets 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.

14.2 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 fields to be filtered.
- Interface identifier.
- Input/output designation.
- *Include/exclude/tag* designation.
- Tag value (if designated).

14.2.1 Filter-Item parameters

The following parameters construct a filter-item:

- Address Type: *source or destination*.
- Tag: *MAC address*.
- Address Mask: *Hex-Mask.*

Each filter-item specifies an address type (source or destination) to match the type in a packet with tokens.

The *address mask* is a MAC address in hex comparing the packet's addresses. The mask is applied to the packet's source destination MAC address before comparing it against the specified MAC address.

The mask specifies the bytes to be logically ANDed with the bytes in the MAC address. This must be the same length as the specified MAC address.

14.2.2 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 the filter item was added to the list. If any of the filter items in the list return TRUE (the rule included in the filter item is applicable to the packet), then the filter-list returns its designated action (*include*, *exclude* or *tag*).

14.2.3 Filter parameters

The following parameters are used to construct a filter:

- Filter-list Names: *ASCII-string, ..., ASCII string*.
- Interface Identifier.
- Filter Address: *input* or *output*.
- Default Action: *include, exclude*, or *tag*.
- Default Tag.

A filter is constructed by associating a group of filter names with an interface and assigning input or output designation. Applying a filter to a packet means each of the associated filter-lists should be applied to received (input) or sent (output) packets on the specified interface.

When a filter evaluates a packet to *include*, the packet is forwarded. When a filter evaluates a packet to *exclude*, it is dropped. When a filter evaluates to a *tag* condition, the packet is forwarded with a tag.

An additional parameter for each filter is the default. This is the result of non-match for all its filter-lists. Default is *include*. Said action can be *include*, *exclude*, or *tag*. If default is *tag*, a *tag* value is given.

14.3 Using MAC filtering tags

MAC Address filtering is handled as 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.

Bandwidth Reservation classification complying with these filters is supported in ATM, Frame Relay, PPP and IP Tunnel interfaces.

Tagging is done by creating a filter item at the MAC filtering configuration prompt and assigning a tag to it. Said tag is used to set up a bandwidth class for all packets associated with it. Tag values must be in the range of 1 to 64.

Supports applying tags to bridged packets only and allows only the packet's MAC address fields to be used in applying the tag.

For further information on using tags in bandwidth reservation, please see manual bintec *Dm715-I Bandwidth Reservation System.*

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 theMAC address filtering tagging feature.

Chapter 15 Configuration and Monitoring MAC Filtering

15.1 Accessing the MAC filtering configuration and monitoring menus

Run **feature mac-filtering** (global configuration menu) to access the MAC filter configuration menu.

Example:

```
Config>feature mac-filtering

-- MAC Filtering user configuration --
Filter Config>
```

Run **feature mac-filtering** (global monitoring menu) to access the MAC filter monitoring menu.

Example:

```
+feature mac-filtering


-- MAC Filtering user console --

Filter+
```

15.2 MAC filtering configuration commands

This section describes the MAC filtering configuration commands. *MAC Filtering Configurations Commands.* on page 148 lists the commands.

MAC Filtering Configurations Commands.

Command	Function
? (HELP)	Displays the available commands or their options.
ATTACH	Adds a filter-list to a filter.
CREATE	Creates a filter-list or filter.
DEFAULT	Sets the default action for a filter.
DELETE	Deletes a previously created filter-list or a filter.
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	Displays information relative to the created filters.
MOVE	Reorders the filter-lists attached to a specific filter.
REINIT	Re-initializes all MAC filtering system without affecting the rest of the router.
SET-CACHE	Changes the cache size for a filter.
UPDATE	Accesses the configuration menu for a filter-list to add or delete filter items on it.
EXIT	Exits MAC filtering configuration menu.

15.2.1 ? (HELP)

Lists the available commands or their options.

Syntax:

```
Filter Config>?
```

Example:

```
Filter Config> ?
  attach       Add a filter list to a filter
  create       Create a filter list or a filter
  default      Set the default action for a filter
```

```
  delete       Delete a command
  detach       Delete a filter-list name from a filter
  disable      Disable MAC filtering
  enable       Enable MAC filtering
  list         List configuration
  move         Re-order filter-lists attached to a filter
  reinit       Reinitializes the MAC filtering system
  set-cache    Changes the cache size
  update       Enter the update filter-list menu
  exit
Filter config>
```

15.2.2 ATTACH

Adds a filter-list to a filter. A filter is constructed by associating a group of filter-lists with an interface. A filter-list is built from one or more filter items.

Syntax:

```
Filter Config>attach <filter-list-name> <filter-number>
  <1..16 chars>    Filter list name
  <1..2147483647>  Filter number
```

filter-list-name	Filter-list identifier.
filter-number	Filter identifier.

Example:

Adds the filter-list “paris” (with identifier 1) to the filter.

```
Filter Config>attach paris 1
Filter Config>
```

15.2.3 CREATE

Creates a filter-list or a filter.

Syntax:

```
Filter Config> create ?
  filter    Create a filter
  list      Create a filter list
Filter config>
```

15.2.3.1 CREATE FILTER

Creates an input or an output filter, associating it with an interface.

Syntax:

```
Filter Config>create filter [input,output] <interface>
```

input	Created filter is applied to input packets at the interface.
output	Created filter is applied to output packets at the interface.
interface	Interface where the filter is applied.

Example:

Creating an input filter on the ethernet0/1 interface.

```
Filter Config>create filter input ethernet0/1
Filter Config>
```

15.2.3.2 CREATE LIST

Creates a filter-list. This list is identified with a name (*Filter-list-name*), a unique string of up to 16 characters.

Syntax:

```
Filter Config>create list <Filter-list-name(1..16 chars)>
```

Example:

Creating a filter-list named “probe-list”.

```
Filter Config>create list probe-list
Filter Config>
```

15.2.4 DEFAULT

Sets the default action for the filter. If none of the filter-lists making up the filter apply to a packet, the default action is executed.

Syntax:

```
Filter config>default ?
  exclude    Set exclude the default action for a filter
  include    Set include the default action for a filter
  tag        Set tag the default action for a filter
Filter config>
```

15.2.4.1 DEFAULT EXCLUDE

Sets the default action for a particular filter to *exclude*.

Syntax:

```
Filter Config>default exclude <filter number (1..2147483647)>
```

Example:

```
Filter Config>default exclude 2
Filter Config>
```

15.2.4.2 DEFAULT INCLUDE

Sets the default action for a particular filter to *include*.

Syntax:

```
Filter Config>default include <filter number(1..2147483647)>
```

Example:

```
Filter Config>default include 3
Filter Config>
```

15.2.4.3 DEFAULT TAG

Sets the default action for a particular filter to *tag* and configures the default tag value.

Syntax:

```
Filter Config>default tag <tag value(1..64)> <filter number(1..2147483647)>
```

Example:

Indicates that filter 1 must tag by default, using tag 4.

```
Filter Config>default tag 4 1
Filter Config>
```

15.2.5 DELETE

Deletes a previously created filter-list or filter.

Syntax:

```
Filter Config>delete ?
  filter    Delete a filter
  list      Delete a filter-list
Filter config>delete
```

15.2.5.1 DELETE FILTER

Deletes a filter.

Syntax:

```
Filter config>delete filter <filter number (1..2147483647)>
```

Example:

```
Filter Config>delete filter 1
Filter Config>
```

15.2.5.2 DELETE LIST

Removes all information associated with a filter-list, including the filter items making up the list. If the filter-list is associated with a created filter, an error message is displayed and nothing is deleted.

Syntax:

```
Filter config>delete list <Filter list name (1..16 chars)>
```

Example 1:

Attempt to delete filter-list1, associated with a filter.

```
Filter Config>delete list filter-list1
CLI Error: Filter-list 'filter-list1' is still attached to filter 1
CLI Error: Command error
Filter Config>
```

Example 2:

Deleting the probe-list, which is not associated with a filter.

```
Filter Config>delete list probe-list
Filter Config>
```

15.2.6 DETACH

Detaches a filter-list name from a filter.

Syntax:

```
Filter Config>detach <filter-list-name> <filter-number>
```

filter-list-name	Filter-list identifier.
filter-number	Filter identifier.

Example:

Detaches the filter-list “paris” (identifier 1) from the filter.

```
Filter Config>detach paris 1
Filter Config>
```

15.2.7 DISABLE

Disables all MAC filtering or a particular filter.

Syntax:

```
Filter Config> disable ?
  all       Disable MAC filtering
  filter    Disable a particular filter
Filter config>
```

15.2.7.1 DISABLE ALL

Disables all MAC filtering.

Example:

```
Filter Config>disable all
Filter Config>
```

15.2.7.2 DISABLE FILTER

Disables a particular filter.

Syntax:

```
Filter config>disable filter <Filter number(1..2147483647)>
```

filter number	Filter identifier. This is obtained by running **list filters**.

Example:

```
Filter Config>disable filter 2
Filter Config>
```

15.2.8 ENABLE

Enables all MAC filtering or a particular filter.

Syntax:

```
Filter config>enable ?
  all       Enable MAC filtering
  filter    Enable a particular filter
Filter config>
```

15.2.8.1 ENABLE ALL

Enables all MAC filtering.

Syntax:

```
Filter Config>enable all
Filter Config>
```

15.2.8.2 ENABLE FILTER

Enables a particular filter.

Syntax:

```
Filter config>enable filter <Filter number(1..2147483647)>
```

filter number	Filter identifier. This is obtained by running **list filter**.

Example:

```
Filter Config>enable filter 1
Filter Config>
```

15.2.9 LIST

Displays information relative to the created filters.

Syntax:

```
Filter Config>list ?
  all       List all configuration
  filter    List a particular filter configuration
Filter config>
```

15.2.9.1 LIST ALL

Lists all configured filter-lists and filters. A list of all filter-lists attached to a filter is not given. Other information displayed includes:

- MAC global filtering state: enabled or disabled.
- Lists of configured filters and the action associated with each.

- Configured filters.

The following information is also displayed for each filter:

- Filter Identifier.
- Filter default action (*tag, include, exclude*). Where default is *tag*, the *tag* is displayed in parenthesis.
- Filter state (enable, disable).
- Interface this is applied to.
- Filter address (input, output).
- Cache size.

Example:

```
Filter Config>list all
Filtering: enabled
Filter List                  Action
-----------                  ------
paris                        INCLUDE
mac-filter                   TAG( 1)
newyork                      EXCLUDE

Filters
-------
Id   Default  State         Ifc            Dir       Cache
--   -------  -----         ---            ---       ------
1    INCLUDE  disabled      ethernet0/0    output    16
2    EXCLUDE  enabled       token-ring3/0  input     16
3    TAG( 3)  enabled       ethernet0/1    input     16
Filter Config>
```

15.2.9.2 LIST FILTER

Displays a list of attached filter-lists for a specified filter and all information for said filter.

Syntax:

```
Filter Config>list filter <Filter Number(1..2147483647)>
```

Example:

```
Filter Config>list filter 1
Id   Default  State         Ifc            Dir       Cache
--   -------  -----         ---            ---       ------
1    INCLUDE  disabled      ethernet0/0    output    16

Filter List                  Action
-----------                  ------
paris                        INCLUDE
Filter Config>
```

15.2.10 MOVE

Re-orders the filter-lists attached to a specified filter.

Syntax:

```
Filter Config>move <filter-list-name1> <filter-list-name2> <filter-number>
```

filter-list-name1,2	Identifiers for the lists to be moved. The list, identified by the *filter-list.name1* parameter, is moved immediately before the list identified by *filter-list-name2*.
filter-number	Filter identifier.

Example:

```
Filter config>list filter 3
Id   Default  State         Ifc            Dir       Cache
--   -------  -----         ---            ---       ------
3    INCLUDE  enabled       ethernet0/0    output    16

Filter List                  Action
```

```
-----------                    ------
rome                           INCLUDE
paris                          INCLUDE
newyork                        INCLUDE
Filter config>move newyork rome 3
Filter config>list filter 3
Id   Default  State         Ifc            Dir       Cache
--   -------  -----         ---            ---       ------
3    INCLUDE  enabled       ethernet0/0    output    16

Filter List                    Action
-----------                    ------
newyork                        INCLUDE
rome                           INCLUDE
paris                          INCLUDE
Filter config>
```

15.2.11 REINIT

Reinitializes all the MAC filtering system (from an existing configuration) without rebooting the entire device.

Syntax:

```
Filter Config>reinit
```

15.2.12 SET-CACHE

Changes the cache size associated with a filter.

Syntax:

```
Filter Config>set-cache <filter-number> <cache-size>
```

filter-number	Filter identifier.
cache-size	Cache size: the number of known entries in the cache associated with the filter. Values range from 4 to 32768. Default is 16.

Example:

```
Filter Config>set-cache 1 32
Filter Config>
```

15.2.13 UPDATE

Accesses the filter-list configuration menu, to add or delete filter items from it. For a description of the commands, please see *MAC filtering list configuration commands* on page 158.

The order the filter-items are specified for a filter-list is important, as it determines the order the filter-items are applied to a packet.

Syntax:

```
Filter Config>update <filter-list-name>
```

Example:

```
Filter Config>update probe
Filter 'probe' Config>
```

15.2.14 EXIT

Exits MAC filtering configuration menu.

Syntax:

```
Filter Config>exit
```

Example:

```
Filter Config>exit
```

```
Config>
```

15.3 MAC Filtering monitoring commands

This section describes the MAC filtering monitoring commands. *MAC Filtering Commands* on page 155 lists the commands.

MAC Filtering Commands

Command	Function
? (HELP)	Displays available commands or options.
CLEAR	Clears statistics.
DISABLE	Disables MAC filtering globally or on a per filter basis.
ENABLE	Enables MAC filtering globally or on a per filter basis.
LIST	Displays information relative to the active filters.
REINIT	Re-initializes MAC filtering system without affecting the rest of the router.
EXIT	Exits MAC filtering monitoring menu.

15.3.1 ? (HELP)

Lists available commands or options.

Syntax:

```
Filter>?
```

Example:

```
Filter+?
  clear      Clears statistics
  disable    Disables MAC filtering
  enable     Enables MAC filtering
  list       Displays MAC filtering configuration
  reinit     Reinitializes the MAC filtering system
  exit
```

15.3.2 CLEAR

Clears all statistics.

Syntax:

```
Filter+clear ?
  all       Clears all statistics
  filter    Clears per filter statistics
```

15.3.2.1 CLEAR ALL

Clears statistics for all filters and filter-lists.

Example:

```
Filter+clear all
Filter+
```

15.3.2.2 CLEAR FILTER

Clears statistics associated with a specified filter and clears all the filter-lists statistics (for said filter).

Syntax:

```
Filter+clear filter <filter-id>
```

Example:

```
Filter+clear filter 1
Filter>
```

15.3.3 DISABLE

Disables all MAC filtering or disables a particular filter.

Syntax:

```
Filter+disable ?
  all       Disables MAC filtering completely
  filter    Disables a specific MAC filter
```

15.3.3.1 DISABLE ALL

Disables all MAC filtering.

Example:

```
Filter+disable all
Filter+
```

15.3.3.2 DISABLE FILTER

Disables a particular filter.

Syntax:

```
Filter+disable filter <filter-id>
```

Example:

```
Filter+disable filter 2
Filter+
```

15.3.4 ENABLE

Enables all MAC filtering or enables a particular filter.

Syntax:

```
Filter+enable ?
  all       Enables MAC filtering completely
  filter    Enables a specific MAC filter
```

15.3.4.1 ENABLE ALL

Enables all MAC filtering.

Example:

```
Filter+enable all
Filter+
```

15.3.4.2 ENABLE FILTER

Enables a particular filter.

Syntax:

```
Filter+enable filter <filter-id>
```

Example:

```
Filter+enable filter 1
Filter+
```

15.3.5 LIST

Displays information on the active filters.

Syntax:

```
Filter+list ?
  all       Displays a summary of the MAC filters configured
  filter    Displays a specific MAC filter configuration
```

15.3.5.1 LIST ALL

Lists all the configured filter-lists and filters (it does not give a list of all filter-lists attached to a filter).

The following information is displayed for each filter:

- Filter identifier.
- Filter default action: *include*, *exclude* or *tag*. When default is *tag*, said *tag* is shown in parenthesis.
- Filter state: enabled or disabled.
- Interface where the filter is applied.
- Filter address: input or output.
- Cache size.
- Number of times a packet has been filtered for already being in the cache associated with the filter.
- Number of packets included by the filter.
- Number of packets excluded by the filter.
- Number of packets tagged by the filter.

Example:

```
Filter+list all
MAC Filtering: enabled
Id Default State Interface      Dir Cache  Hit%   Inc       Exc       Tag
-- ------- ----- -------------- --- -----  ----   ---       ---       ---
1  EXCLUDE ENA   token-ring3/0  IN  32     100.0  0         0         0
2  TAG( 3) DIS   serial0/1      IN  16     100.0  0         0         0
3  INCLUDE ENA   ethernet0/0    OUT 16     100.0  0         0         0
4  INCLUDE ENA   bri0/0         OUT 16     100.0  0         0         0
Filter+
```

15.3.5.2 LIST FILTER

Displays a list of attached filter-lists for the specified filter and all subsequent information for said filter.

The information shown for the filter is the same as that shown through the list all command. The following information is shown for each filter-list:

- Memory in bytes, occupied by the filter-list control structure.
- Action associated with the filter-list: *include*, *exclude* or *tag*.
- Filter-list identifier (name).
- Number of times this filter-list has been used.

Example:

```
Filter+list filter 1
Id Default State Interface      Dir Cache  Hit%   Inc       Exc       Tag
-- ------- ----- -------------- --- -----  ----   ---       ---       ---
1  EXCLUDE ENA   token-ring3/0  IN  32     100.0  0         0         0

Filter Lists:
MemUse     Action    Name                    Count
------     ------    ----                    -----
92         TAG( 1)   mac-filter              0
Filter+
```

15.3.6 REINIT

Reinitializes the entire MAC filtering system (from an existing configuration) without rebooting.

Syntax:

```
Filter+reinit
```

15.3.7 EXIT

Exits the MAC filter monitoring menu.

Syntax:

```
Filter+exit
```

Example:

```
Filter+exit
+
```

15.4 MAC filtering list configuration commands

This section describes the MAC filter-list configuration commands. *MAC Filtering Update Commands* on page 158 shows the MAC filter-list configuration commands.

MAC Filtering Update Commands

Command	Function
? (HELP)	Displays available commands or options.
ADD	Adds a filter item to a configured filter-list.
DELETE	Removes filter-items from a filter-list.
LIST	Displays the filter-list configuration.
MOVE	Reorders the filter-lists attached to a specified filter.
SET-ACTION	Configures the action to be executed by the filter-list.
EXIT	Exits the MAC filter-list configuration menu.

15.4.1 ? (HELP)

Lists the available commands or their options.

Syntax:

```
Filter 'filter-list-name' config>?
```

Example:

```
-- MAC Filtering list configuration --
Filter 'probe' config>?
  add           Add a filter-item to a filter-list
  delete        Delete a filter-item from a filter-list
  list          List configuration
  move          Re-order filter-items within the filter-list
  set-action    Set filter list action
  exit
Filter 'probe' config>
```

15.4.2 ADD

Adds filter-items to a filter-list. This adds a hexadecimal number to compare against the source or destination MAC address.

The order you add filter-items to a filter-list is important, as it determines the order the filter-items are applied to a packet.

The first match that occurs stops the application of filter-items, and the filter-list evaluates to *include, exclude* or *tag*, depending on the designated action of the filter-list.

Syntax:

```
Filter 'filter-list-name' config>add ?
  source         Compare against the source MAC address
        <mac>          MAC pattern to compare
               <mac-mask>     MAC mask
  destination    Compare against the source MAC address
        <mac>          MAC pattern to compare
```

```
               <mac-mask>     MAC mask
Filter 'filter-list-name' config>
```

source	Filter item through source MAC address. The packet source MAC address is used to compare with the pattern.
destination	Filter item through destination MAC address. The packet destination MAC address is used to compare with the pattern.
mac	MAC address used as a pattern to compare with the corresponding packet MAC address. The MAC address can be entered in canonic format (entering the characters without separations or separating them with a hyphen) or in non-canonic format (entering the characters, separating them with a colon).
mac-mask	Mask applied to the address corresponding to the packet before being compared with the pattern MAC address. Said mask is applied through the AND logical operation and must be the same length as the MAC address.

Example 1:

Add a filter item to filter packets with source address 00-A0-26-00-AC-5x, where x can be any value.

```
Filter 'probe' config>add source 00-a0-26-00-ac-50 ff-ff-ff-ff-ff-f0
Filter 'probe' config>
```

Example 2:

Add a filter item to filter packets with destination address 00:00:C9:09:66:49.

```
Filter 'probe' config>add destination 00:00:c9:09:66:49 ff:ff:ff:ff:ff:ff
Filter 'probe' config>
```

15.4.3 DELETE

Removes filter-items from a filter-list. Delete filter items by specifying the filter-item-number assigned to the item. Run **list** to check the number assigned to each filter item.

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 'probe' config>list canonical
Action: INCLUDE
Id  Type  MAC Address                Mask
--  ----  ------------               ----
1   SRC   00-a0-20-33-11-22          ff-ff-ff-ff-ff-ff
2   SRC   00-a0-26-00-ac-50          ff-ff-ff-ff-ff-f0
3   DST   00-00-93-90-66-92          ff-ff-ff-ff-ff-ff
Filter 'probe' config>delete 1
Filter 'probe' config>list canonical
Action: INCLUDE
Id  Type  MAC Address                Mask
--  ----  ------------               ----
1   SRC   00-a0-26-00-ac-50          ff-ff-ff-ff-ff-f0
2   DST   00-00-93-90-66-92          ff-ff-ff-ff-ff-ff
```

15.4.4 LIST

Displays the filter-list configuration. This shows the following information on 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: *include*, *exclude* or *tag*. Where the action is *tag*, the *tag* associated with the filter-list is displayed between parenthesis ( ).

Syntax:

```
Filter 'filter-list-name' config>list ?
  canonical
  noncanonical
Filter 'filter-list-name' config>
```

15.4.4.1 LIST CANONICAL

Displays the filter-list configuration, showing MAC addresses in canonical format.

Example:

```
Filter 'probe' config>list canonical
Action: INCLUDE
 Id  Type  MAC Address                Mask
--  ----  ------------               ----
1   SRC   00-a0-26-00-ac-50          ff-ff-ff-ff-ff-f0
2   DST   00-00-93-90-66-92          ff-ff-ff-ff-ff-ff
Filter 'probe' config>
```

15.4.4.2 LIST NONCANONICAL

Displays the filter-list configuration, showing MAC addresses in non-canonical format.

Example:

```
Filter 'probe' config>list noncanonical
Action: INCLUDE
Id  Type  MAC Address                Mask
--  ----  ------------               ----
1   SRC   00:05:64:00:35:0a          ff:ff:ff:ff:ff:0f
2   DST   00:00:c9:09:66:49          ff:ff:ff:ff:ff:ff
Filter 'probe' config>list noncanonical
```

15.4.5 MOVE

Re-orders filter-items within the filter-list. The filter-item (number is specified by *filter-item-name 1*) is moved and re-numbered so it's inserted just before *filter-item-name 2* . Use the **list** command to check the number assigned to each filter item.

Syntax:

```
Filter 'filter-list-name' Config>move <filter-item-name1> <filter-item-name2>
```

Example:

```
Filter 'probe' config>list canonical
Action: INCLUDE
Id  Type  MAC Address                Mask
--  ----  ------------               ----
1   SRC   00-a0-26-00-ac-50          ff-ff-ff-ff-ff-f0
2   DST   00-00-93-90-66-92          ff-ff-ff-ff-ff-ff
3   SRC   00-a0-26-00-aa-23          ff-ff-ff-ff-ff-ff
Filter 'probe' config>move 3 2
Filter 'probe' config>list canonical
Action: INCLUDE
Id  Type  MAC Address                Mask
--  ----  ------------               ----
1   SRC   00-a0-26-00-ac-50          ff-ff-ff-ff-ff-f0
2   SRC   00-a0-26-00-aa-23          ff-ff-ff-ff-ff-ff
3   DST   00-00-93-90-66-92          ff-ff-ff-ff-ff-ff
Filter 'probe' config>
```

15.4.6 SET-ACTION

Sets the action to be executed by the filter-list. If one of the filter-items on the filter-list matches the contents of the packet being considered for filtering, the filter-list evaluates to this condition. Default is include.

Syntax:

```
Filter 'filter-list-name' config>set-action ?
  exclude    Set exclude the action for the list
  include    Set include the action for the list
  tag        Set tag the action for the list
        <tag-value>           tag value
Filter 'filter-list-name' config>
```

include	Action to be executed by the filter-list is *include*: if the packet coincides with a filter item, said packet is not dropped.
exclude	Action to be executed by the filter-list is *exclude*: if the packet coincides with a filter item, said packet is dropped.
tag	Action to be executed by the filter-list is *tag*: if the packet coincides with a filter item, a tag is assigned to the packet.
tag-value	Value between 1 and 64, for the tag to be assigned to a packet where the action to execute is *tag*.

Example:

```
Filter 'probe' config>set-action tag 1
Filter 'probe' config>
```

15.4.7 EXIT

Exits the MAC filter-list configuration menu.

Syntax:

```
Filter 'filter-list-name' config>exit
```

Example:

```
Filter 'probe' config>exit
Filter config>
```

Chapter 16 Using Protocol Threading Through a Bridged Network

16.1 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.

16.2 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 MAC address, the end station transmits an arp request 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 and send it over their connected networks.

(3) As the arp request 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 compiles a list of bridge and segment number pairs describing the path to the destination.

(4) When the arp request 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 them.

(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.

16.3 DNA threading

Digital Network Architecture (DNA) end stations use ARE (All Routes Explore) to discover a route. Both DNA end stations and 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 a **no_route** state. When this occurs the end station sends a 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 compiles 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 said entry, it either updates the RIF (if the previous entry is an **st_route**) or it ignores it. Either way, 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 said STE. When this happens, the entry state changes to **st_route**.

DNA end stations contain an independent RIF timer. When this timer times out 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.

16.4 Apollo threading

Apollo end stations use STE frames to discover a route. Both 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 the destination end station, each bridge (that forwards the packet) adds its own bridge number and segment number to the RIF in the STE. As the frames passes through the bridged network, the RIF compiles a list of bridge and segment number pairs describing the path to the destination.

(5) When the STE 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 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 it. In either 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 said 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 said timer times out 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.

16.5 IPX threading

IPX end stations check each packet they receive for an RIF. If the RIF does not exist in the table, they add it to said table and designate that route as **have_route**. If the RIF indicates 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 times out, the entry in the table is cleared and won't be reentered until another packet arrives containing an RIF for said entry.

16.6 Threading AppleTalk 1 and 2

AppleTalk end stations use ARP and XID packets to discover a route. Both 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 compiles 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 them.

(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 times out, 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.