Chapter 16

Bridging

Introduction

This chapter describes:

- the bridging function on the router
- how bridging is implemented on the router
- how to configure and operate the router in one of the following modes:
    - LAN to LAN local bridging
      This configuration enables multiple LAN segments to be connected together to form an extended LAN within a local environment.
    - LAN to LAN remote bridging
      This configuration enables a LAN to be connected to a wide area network. It is used to extend a LAN across remote sites.
    - VLAN-to-WAN bridging
      This configuration enables traffic from a single VLAN to be shared across a PPP link.The WAN-to-WAN bridging functions and the VLAN-VLAN remote bridging modes cannot operate simultaneously. See "VLAN-to-WAN Bridging" on page 16-7.
    - WAN-to-WAN bridging
      This configuration bridges traffic between two or more ports to link remote WAN connected LANs so as to forward layer two traffic between them. See "WAN-to-WAN Bridging" on page 16-10.

To configure layer 2 frame forwarding between switch ports on a local network, use the VLAN configuration described in Chapter 8, Switching.

The Bridge module provides the following functionality:

- Dynamic configuration via management commands or SNMP requests. Configuration changes within the bridge module take effect immediately without requiring the bridge module or the router to be reset.
- Management of the bridge station map.
- Learning MAC addresses.
- Filtering and forwarding packets. It accepts all packets on its interfaces and filters and forwards them with no distinction between protocols. The bridge module can also be configured to process packets of given protocol types. There are three classes of protocol: those that are routed, those that are bridged, and those that are ignored. This lets the router act as a bridge for only the protocols that must be bridged. This makes the process more efficient for wide area bridging.
- Support for ports over point-to-point (PPP), ATM channels, and Frame Relay interfaces.
- Support for on-demand ports over PPP interfaces.
- Support for the bridge MIB (RFC 1493), the Internet Draft for the Point-to-Point Protocol Bridging Control Protocol, and RFC 1490 "*Multiprotocol Interconnect over Frame Relay*".
- Operation of the spanning tree algorithm.
- Operation of configurable bridge filters that further modify the filtering and forwarding processes.

Attachment to a VLAN to provide Remote Bridging.

Bridging on the Router

The implementation of the bridge module in the router follows the IEEE 802.1D-1990 Standard, "*Media Access Control (MAC) Bridges*".

The IEEE Standard 802.1G "*Remote MAC Bridging*" does not specify any mechanisms (protocols, procedures, communication technologies, etc.) for transporting frames between remote bridges over virtual ports. The implementation of the bridge module in the router follows the IEEE 802.1D-1990 Standard, "*Media Access Control (MAC) Bridges*". The standards supported by the switch for providing such transport are: RFC1171 "*Point-to-Point Protocol for the Transmission of Multi-Protocol Datagrams Over Point to Point Links,*" and RFC 1490 "Multiprotocol Interconnect over Frame Relay."WAN ports are either Frame Relay interfaces, as specified in RFC1490; or PPP interfaces, as specified in RFC 2878.

A bridge that has one or more interfaces connected to a wide area network, is called a *remote bridge*. These bridges can be used to form extended LANs across a wide area network. A particular adaptation of this device, called a *VLAN-to-WAN bridge,* enables terminals attached to a single VLAN to connect to remote terminals across a PPP or Frame Relay wide area network. Note however, that WAN-to-WAN bridging and VLAN-to-WAN bridging cannot simultaneously operate within the same device.

For more information, see the sections: "Internal Representation of the WAN-to-WAN Bridge" on page 16-11, and "WAN-to-WAN Bridging" on page 16-10.

A bridge is also referred to as a MAC (media access control) bridge, a data link relay or a layer 2 relay. A bridge connects multiple LAN segments together to form an extended LAN. This enables stations connected on different LANs to communicate with one another as if they were on the same LAN. Because bridges function at the data link layer, transferring data link frames between their attached networks; they are able to operate independently of the higher layer protocols.

A bridge accesses each physical link according to the rules for that particular network. Because access is not always instant, a bridge must be capable of storing and forwarding frames. Having this store and forward functionality also enables it to examine each frame's destination address field and forward the frame to the network containing the appropriate station. In this way, a bridge can act as an intelligent filtering device, redirecting or blocking the movement of frames between networks.

Remote Bridging

Network bridging originally developed as a way to extend boundaries of local area network connections. However, the increasing need to interconnect remotely located LANs has resulted in two different technology directions.

- To extend layer two connectivity, enabling it to interface to wide area networks (WANs) - Remote Bridging/Layer Two Switching.
- To connect remote LANS at the layer three utilising the internet protocol (IP) or similar - Network Routing/Layer Three Switching.

Two remote bridge configurations are supported by the switch, VLAN-to-WAN bridging, and WAN-to-WAN bridging. In order to provide these configurations, the switch contains both a layer two switch and a bridge module. Certain layer two functions are therefore configured using the switch and VLAN commands, while other functions use the bridge module commands. The internal structure of the two configuration modes, VLAN-to WAN-and WAN-to-WAN, are slightly different.

For more information, see “WAN-to-WAN Bridging” on page 16-10.

Virtual Ports and Switch Ports

Traffic is bridged over two distinct types of ports, switch ports and virtual ports. A switch port presents a MAC and physical level interface to a LAN connected device, and always forms part of a VLAN. A virtual port can be thought of as being a named connection point for a specific inter-bridge communications path over a wide area network. An important concept is that, as *virtual* entities, these ports do not represent physical connections, and that a single physical interface may have multiple virtual ports assigned to it.

The bridge supports virtual port connectivity via either Frame Relay, ATM, or point-to-point (PPP) network interfaces. When using Frame Relay, each virtual port is mapped to a Frame Relay virtual circuit.

The **port** parameter is used in VLAN commands to specify the switch ports (see Chapter 8, Switching*).* The **port** parameter used in BRIDGE commands refers to virtual ports. An individual virtual port number is associated with each link, ATM channel, or a Frame Relay virtual circuit.

Each port is uniquely identified by a port number. A bridge port may be in one of the states in the following table (Table 16-1).

Table 16-1: Bridge port states

State	Meaning
DISABLED	Bridging operations are disabled on the port. In particular, the Forwarding Process and the Spanning Tree entity are disabled for transmit and receive operations on the port.
LISTENING	The port is enabled for receiving frames.
LEARNING	The port is enabled for receiving frames, and the Forwarding Process is placing new source address information in the station map.
FORWARDING	The normal state for a bridge port. The Forwarding Process and the Spanning Tree entity are enabled for transmit and receive operations on the port.
BLOCKING	The Spanning Tree entity has disabled the Forwarding process for transmit and receive operations on the port, but the Spanning Tree entity itself remains enabled for transmit and receive operations on the port.

A bridge port may be configured over a PPP interface with the IDLE parameter set (non-zero). In this case the PPP link is active when traffic is sent, timing out after the idle period. This is referred to as *on-demand* bridging. On-demand bridging is particularly useful when the PPP interface is associated with an ISDN call. The ISDN call is automatically activated when the bridge module transmits packets over the link, and deactivated when the idle timer expires after a set period of inactivity. On-demand bridge links are not supported when the Spanning Tree Protocol is enabled.

Broadcast or multicast packets can repeatedly activate an on-demand link, resulting in unnecessary call charges. Because of this, on-demand bridging may not be suitable in some situations and in other situations the router may need careful configuration (the addition of filters, for example) to avoid unnecessary calls.

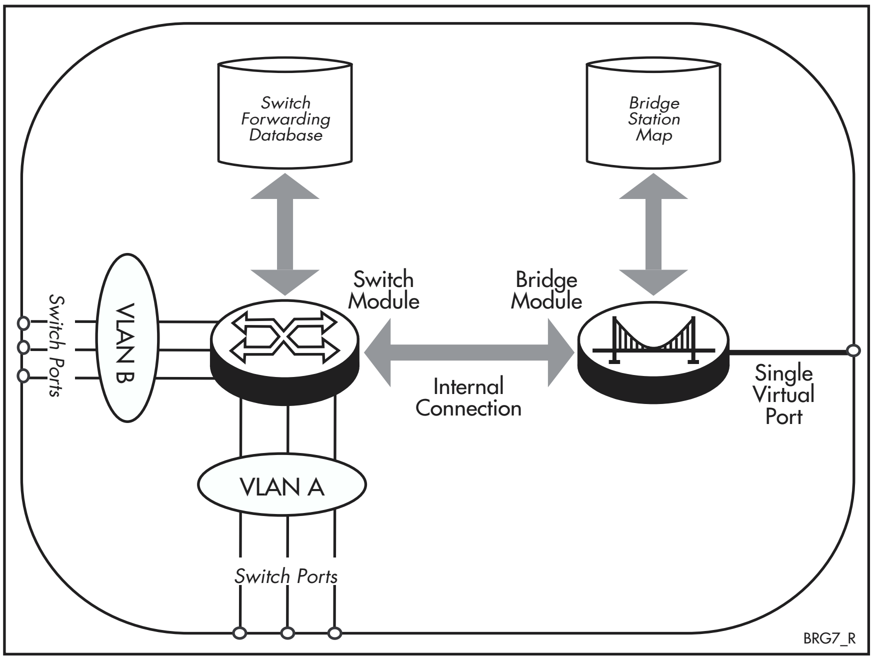
VLAN-to-WAN Bridging

In general, it is better to route a protocol than to bridge it. However, sometimes bridging is a more appropriate solution, particularly where unroutable upper layer protocols are used. These protocols sometimes produce high levels of broadcast messages that can overload a network, an effect that gets progressively worse as the number of devices increases. Although this situation may not pose a problem to the high bandwidths available on local area networks, it could heavily congest the more limited bandwidths available on wide area links. This effect can be reduced by adding a VLAN to a bridge and then limiting the VLAN to devices that require wide area connections. The remote VLAN bridge is able to support up to 16 VLANs

Internal Representation of the VLAN-to-WAN Bridge

Figure 16-1 on page 16-7 shows an internal representation of the VLAN-to-WAN bridge configuration. The bridge and switch symbols are internal functional representations and are not standalone devices.

Figure 16-1: Internal representation of the VLAN-to-WAN bridge



Local ports connect to the switch module as VLAN members and provide layer two connectivity for their attached terminals. An internal data path (shown by the horizontal grey arrow) provides connectivity between the two modules. All station address learning is achieved using the switch's forwarding database.

The VLAN-to-WAN Bridging Process

The switch module provides layer two connectivity for locally attached ports within the same VLAN. An internal data connection, shown by the horizontal grey arrow in Figure 16-1 on page 16-7 , provides connectivity between the two modules. The *switch forwarding database*, and the *bridge station map* both employ their own respective processes to learn the addresses of their attached devices.

Local devices within the same VLAN, utilise the forwarding database and their traffic is switched by hardware at layer two. Remote devices within the same VLAN, utilise both the forwarding database and the station map, hence their traffic is *remote bridged* across the wide area link.

The switch module examines the MAC addresses of frames received on its switch ports. If it recognises an address as belonging to a locally attached device, it forwards the frame to the appropriate port. If it recognises a frame's MAC address as belonging to a device residing over the wide area link, it forwards the frame to the bridge module via the internal connection shown. The bridge module then forwards the frame across the wide area link.

VLAN-to-WAN data always crosses the wide area link as tagged frames. At the link's remote end, the bridge may either retain the VID tags for forwarding the frames to a tagged port, or remove the VID tags for forwarding the frames to an untagged port within the VLAN. For this reason you must set the **set bridge stripvlantag** command on page 16-48 to **off**.

LAN-to-LAN Bridging

Bridges are used to:

- Increase the physical extent and/or the maximum number of stations on a LAN.

  LANs are limited in their physical extent by the signal distortion and propagation delay characteristics of the media. A bridge overcomes this limitation by receiving a frame on one LAN and then retransmitting the frame on another LAN, using the normal access methods for each LAN. The physical characteristics of the LAN media also place a practical limit on the number of stations that can be connected to a single LAN segment. A bridge overcomes this limitation by joining LAN segments together to form an extended LAN capable of supporting more stations than either of the individual LANs.

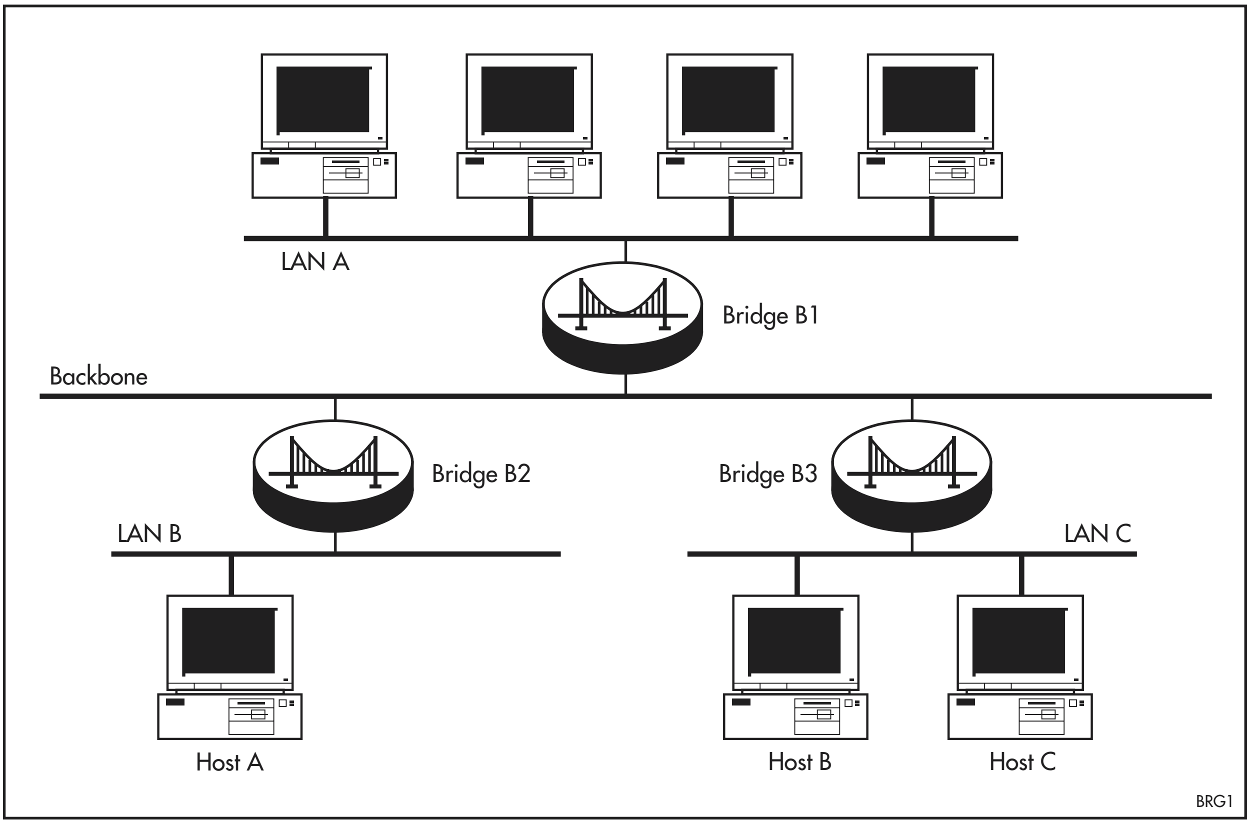
- Connect LANs that have a common data link layer protocol but different physical media, for example, Ethernet 10BASE5, 10BASE2 and 10BASEF.
- Increase the availability of LANs by allowing multiple redundant paths to be physically configured, and selected dynamically, using the Spanning Tree algorithm.
- Reduce the load on a LAN or increase the effective bandwidth of a LAN, by filtering traffic.

  A typical scenario is a campus network comprising multiple Ethernet LANs, one of which is designated the *backbone* LAN (Figure 16-2 on page 16-9). The only devices attached to the backbone are bridges, and perhaps a router for connection to a wide area network. All the other LANs have multiple stations (e.g. PCs, workstations, file servers, print servers, etc.) and are connected to the backbone via the bridges.

  The bridges filter LAN traffic, ensuring that traffic between stations on the same LAN is not passed to the backbone; only traffic intended for a station on another LAN is passed to the backbone; and only traffic intended for a station on the local LAN is received from the backbone.

In Figure 16-2 on page 16-9, Bridge B3 confines traffic between hosts B and C to LAN C, while traffic between hosts A and B is passed to the backbone. Bridge B1 prevents traffic between hosts A and B from passing to LAN A.

Figure 16-2: An example bridged LAN



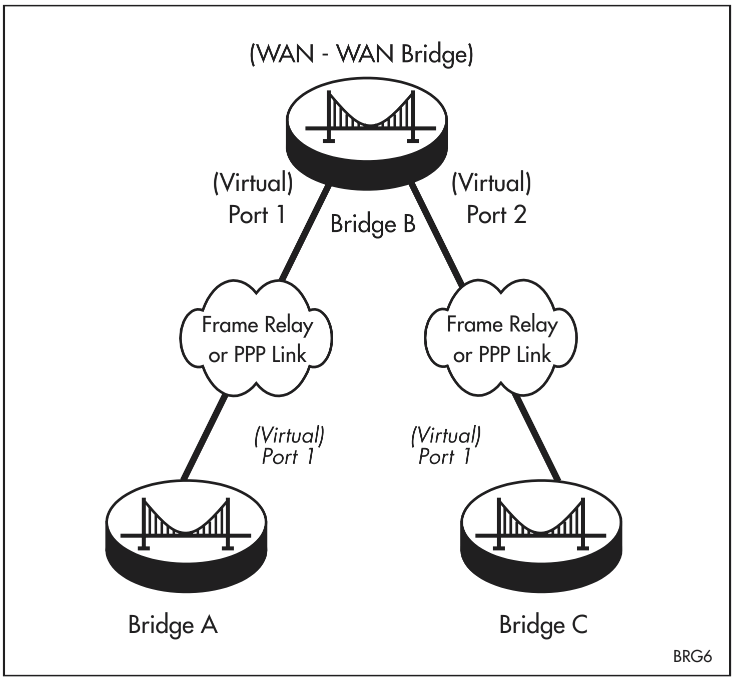
The router can bridge between fixed ports and ports in expansion options, but cannot bridge from one VLAN to another VLAN.

WAN-to-WAN Bridging

This configuration is used to forward frames between two or more LANs connected via a wide area network and using ATM, Frame Relay or PPP. In this configuration the bridge acts simply as a layer two forwarding device and is unable to forward traffic from the wide area network to its own LAN ports.

Figure 16-3 on page 16-10 below shows a possible WAN-to-WAN bridge configuration.

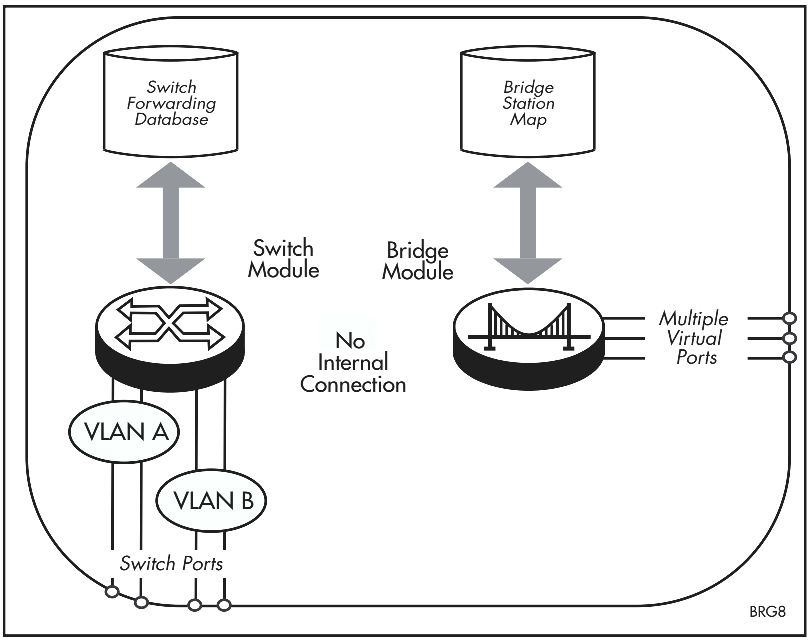
Figure 16-3: Example configuration of a WAN-to-WAN bridge



Internal Representation of the WAN-to-WAN Bridge

Figure 16-4 on page 16-11 below shows an internal representation of the VLAN-to-WAN bridge configuration. The bridge and switch symbols are internal functional representations, and do not represent stand alone devices.

Figure 16-4: Internal representation of WAN-to-WAN bridge



Local ports connect to the switch module as VLAN members and provide layer 2 connectivity for their attached terminals. WAN connected virtual ports connect to a bridge module to provide WAN-to-WAN bridging. Accordingly, station learning occurs in two separate locations: the Bridge Station Map for the virtual (WAN) ports, and the Switch Forwarding Database for the local ports. Unlike the VLAN-to-WAN configuration, there is no communication path between the switch and bridge modules. The LAN switch ports are configured using the switch commands and the WAN virtual ports are configured using the bridge commands.

Bridge Learning and Forwarding

Bridging comprises two separate but related processes - learning and forwarding. Both processes assume that each station on the extended LAN has a unique data link layer address, and that all data link layer frames have a header that includes the source (sender’s) address and destination (recipient’s) address.

Both the learning and the forwarding functions are handled in two areas, depending on the port type. For virtual ports (in the WAN-to-WAN mode) these functions are handled using the bridge station map; and for switch ports (in the VLAN-to-WAN mode) they are handled using the switch forwarding database.

This chapter describes the learning and forwarding applied to the virtual ports, meaning functions that use the bridge station map. For details of learning and forwarding applied to switch ports, see Chapter 8, Switching.

The Learning Process

The learning process uses an *adaptive learning* algorithm, sometimes called *backward learning*, to discover the location of each station on the extended LAN.

The bridge receives every frame on every network to which it is attached, regardless of the frame's destination address. The bridge compares each frame's source address against entries listed in its station map. The station map contains one entry for every unique station address known to the bridge. It also relates each station's (source) address to either a switch port, or a virtual port, on the bridge. Using this information, the bridge determines the port (if any) to transmit frames whose destination address matches the entry in the station map.

If the frame's source address is not already listed in the bridge station map, the address is added, and an aging timer for that entry is started. If the frame's source address is listed, the aging timer for that entry is restarted.

If the aging timer for an entry in the bridge station map expires before another frame with the same source address is received, the entry is removed from the map. This prevents it from filling up with information about stations that are inactive, or have been disconnected from the network, while ensuring that entries for active stations are kept alive.

Learning on a WAN-to-WAN Bridge

The learning process uses an *adaptive learning* algorithm, sometimes called *backward learning*, to discover the location of each station on the extended LAN.

The bridge module receives frames from its virtual ports and compares each frame's source address against entries listed in its bridge station map. This map contains one entry for every unique station known to the bridge. It also relates each station's (source) address to a virtual port on the bridge. Using this information, the bridge determines on which virtual port (if any) to transmit frames whose destination address matches the entry in its bridge station map.

If the frame's source address is not already listed in the bridge station map, the address is added, and an aging timer for that entry is started. If the frame's source address is listed, the aging timer for that entry is restarted.

If the aging timer for an entry in the bridge station map expires before another frame with the same source address is received, the entry is removed from the map. This prevents it from filling up with information about stations that are inactive, or have been disconnected from the network, while ensuring that entries for active stations are kept alive.

The Forwarding Process

The bridge forwards received frames that are to be relayed to other bridge ports, filtering out frames on the basis of information contained in the bridge station map and on the state of the ports.

Forwarding occurs if the port on which the frame was received is in the 'Forwarding' state. The destination address of each frame is then looked up in the bridge station map. If this address is not found, the bridge floods the frame on all ports except the port on which the frame was received. If the address is found, but the entry is marked as 'No Forwarding,' or the indicated port is not in the 'Forwarding' state, or the indicated port is the same as the port on which the frame was received, the frame is discarded. Otherwise, the frame is forwarded to the indicated port.

This whole process can further be modified by the action of bridge filters. These are configurable filters that enable bridged frames to be checked against a number of entries. If a match is made to a filter entry, the port forwarding permissions of the filter are applied in addition to those declared in the bridge station map. Note that in the event of a conflict, the permissions of the filter override those defined in the station map.

Frames that are forwarded over a wide area link can be assigned a priority based on the frame's protocol specifier. The manager can assign priorities to different protocols using the **add bridge protocol** command on page 16-30.

Frames are assigned a priority immediately before being queued for forwarding. All higher priority frames are forwarded before any of a lower priority. Priority levels range from 0 (lowest) to 4 (highest). The default priority is 1.

The router bridges VLAN tagged and untagged frames between Ethernet, Point-to-Point, ATM and Frame Relay interfaces. The maximum allowable frame length for an interface that supports VLAN tagged frames is 1522 bytes. The router checks the size of each frame arriving on the interface. The interface bridges tagged frames up to 1522 bytes and untagged frames up to 1518 bytes. Frames exceeding these respective sizes are discarded.

Forwarding on a WAN-to-WAN Bridge

The bridge forwarding process forwards frames that are to be relayed to other virtual ports, filtering out frames on the basis of information contained in the bridge station map.

The bridge first looks at the source address of each frame it receives and adds each new address into its station map; it then looks up each frames destination address. If a match is found, the frame is forwarded to the appropriate virtual port. If a match is not found, the bridge floods the virtual ports to locate the device whose address matches that contained within the received frame. Once the source address is located, its forwarding details are recorded. Subsequent frames bearing this address can then be forwarded directly to the appropriate virtual port.

This whole process can further be modified by the action of bridge filters. These are configurable filters that enable bridged frames to be checked against a number of entries. If a match is made to a filter entry, the port forwarding permissions of the filter are applied in addition to those declared in the bridge station map. Note that in the event of a conflict, the permissions of the filter override those defined in the station map.

Filtering

The router can function simultaneously as both a bridge and a router. The decision about what data is bridged and routed is based on protocol. All packets of a particular protocol transmitted or received on an interface enabled for bridging may be bridged **or** routed, but not both. By default, all supported protocols are routed, providing that the appropriate routing module has been correctly enabled and configured. Only protocols specifically enabled for bridging are bridged. For example, if an interface on the router has been set to route IP then only non-IP frames would be considered for bridging. This policy limits the potential traffic being transported over slow WAN links, and is the first form of filtering implemented by the bridge module.

The second form of filtering is specified in the IEEE 802.1D Standard "*Media Access Control (MAC) Bridges*". The destination MAC address of a frame to be forwarded is checked against the station map. If there is no entry for the destination address, the frame is transmitted on all ports (except the port on which the frame was received) that are in the 'Forwarding' state. This process is referred to as *flooding*. If an entry is found in the station map is marked as 'No Forwarding' or it points to the same port that the frame was received on, the frame is discarded. Otherwise, the frame is transmitted on the port specified by the entry in the station map.

The station map is stored in RAM and entries are added automatically by the Learning Process. Station map entries may also be added manually by using either the command line interface, or SNMP.

The forwarding process queries the station map as it reads the MAC destination address field of each frame and decides whether to forward the frame on to its appropriate port.

An entry is automatically deleted from the station map when its aging timer expires.

The third form of filtering in the router is by the addition of a configurable bridge filter to a bridge port. Configurable bridge filters are applied when the router is configured for WAN-to-WAN bridging.

Bridge filters consist of a number of entries, each of which contains a match condition and a port list. When a frame is received on a port that has a filter configured to it, the frame is checked against all entries in the filter. If a match is found, the port list is used to modify the ports to which the frame can be forwarded, subject to the information in the station map and the state of the port. The port list can indicate that the frame can be forwarded to all ports, to no ports (in which case the frame is immediately discarded) or to a subset of the configured ports.

Where a filter is applied, frames not meeting the criteria in at least one filter entry are discarded. Therefore, a general filter entry may be included at the end of a filter to ensure that frames not explicitly filtered out are passed on by the filter, if this is required.

Telnet to a Router Bridging IP

A router that is bridging a protocol may not also route that protocol. For a router bridging IP this would mean that an IP interface could not be added and therefore the router could not be managed remotely using Telnet. A special case is, therefore, supported for routers bridging IP. A single IP interface can be added to an Ethernet interface that is bridging IP, to enable the router to be accessed via Telnet. To do this, the IP module should be enabled and a single IP interface created on a LAN. On each router in a bridged IP network, the IP module should be enabled and a single IP interface created on an Ethernet port. This allows Telnet access to remote routers reached via Point-to-Point Protocol, ATM and Frame Relay links as well as local routers.

The configuration must be specifically set to bridge IP and ARP. It is not sufficient to bridge all Ethernet type II packets without specifically bridging IP and ARP packets as well.

Spanning Tree Protocol

A sequence of LANs and bridges may be connected together in an arbitrary physical topology resulting in more than one path between any two bridges. If a loop exists, frames transmitted onto the extended LAN would circulate around the loop indefinitely, decreasing the performance of the extended LAN. On the other hand, multiple paths through the extended LAN provide the opportunity for redundancy and backup in the event of a bridge experiencing a fatal error condition.

The spanning tree algorithm ensures that the extended LAN contains no loops and that all LANs are connected by:

- Detecting the presence of loops and automatically computing a logical loop-free portion of the topology, called a *spanning tree*. The topology is dynamically pruned to a spanning tree by declaring a bridge (or one of its ports) redundant, and placing the bridge (or port) into a ‘No Forwarding’ state.
- Automatically recovering from a bridge failure that would partition the extended LAN by reconfiguring the spanning tree to use redundant bridges.

The logical tree computed by the spanning tree algorithm has the following properties:

- A single bridge, called the *root bridge*, forms a unique root to the tree. The root bridge is the bridge with the lowest bridge ID. Each bridge in an extended LAN is uniquely identified by its bridge ID, which comprises the bridge’s root priority (a spanning tree parameter) and its MAC address.
- Each bridge or LAN in the tree, except the root bridge, has a unique parent.
- The unique parent of a LAN is the *designated bridge* for the LAN. Each LAN has a single bridge, called the designated bridge, that logically connects the LAN to the next LAN closer to the root bridge. Each port connecting a bridge to a LAN has an associated *cost*. The *root path cost* is the sum of the costs for each port between the bridge and the root bridge. The designated

bridge for a LAN is the bridge on the LAN with the lowest root path cost, and therefore logically closer to the root bridge. If two bridges on the same LAN have the same lowest root path cost, the bridge with the lowest bridge ID is elected the designated bridge.

- The unique parent of a bridge is the LAN, to which the bridge is attached, that is closest to the root bridge.

The spanning tree computation is a continuous, distributed process. The algorithm uses the following steps to establish the spanning tree:

1. A unique *root* bridge is elected by the bridges in the extended LAN.
2. A designated bridge is elected for each LAN in the extended LAN by the bridges in the LAN.
3. The logical spanning tree is computed and redundant paths are removed.

Once the spanning tree is established, it is maintained by:

1. Replacing a failed bridge with a redundant backup bridge.
2. Detecting and removing loops by declaring a bridge redundant and removing it from the logical spanning tree.
3. Maintaining address timers that control the aging of FDB address entries.

To implement the spanning tree algorithm, bridges communicate with one another using the Spanning Tree Protocol. The primary protocol data unit (PDU) is the *Hello message* or *Configuration Bridge Protocol Data Unit* (BPDU), which includes the following information:

- The bridge ID of the root bridge.
- The distance (or cost) from this bridge to the root bridge.
- The bridge ID of the designated bridge on this LAN.

Hello messages are initiated a regular intervals by the root bridge and propagate through the extended LAN.

Electing the Root Bridge and Designated Bridge

Each spanning tree has a *root bridge*, which initiates the propagation of Hello messages through the extended LAN, and sets the values of parameters that control the spanning tree computation process. The root bridge is the bridge with the lowest bridge ID and is elected by the exchange of Hello packets. When a bridge receives a Hello packet it compares the value of the root bridge ID in the message to the value of the root bridge ID parameter in its own spanning tree database. If the value in the message is better, the bridge stores the new value in its database and sends Hello messages with the new value out on its other ports. Otherwise, the bridge continues to send out Hello messages with the value currently stored in its spanning tree database. By this process all bridges in the extended LAN eventually learn the bridge ID of the root bridge.

Each LAN has a single bridge, called the *designated bridge*, that logically connects the LAN to the next LAN closer to the root bridge. The designated bridge for a LAN is the bridge on the LAN with the lowest root path cost and bridge ID. The designated bridge is elected by the exchange of Hello messages, in the same way that the root bridge is elected. The election of a new root bridge, or a bridge becoming unavailable due to a fatal error condition, normally results in the election of a new designated bridge in the next few rounds of Hello messages.

Configuration Examples

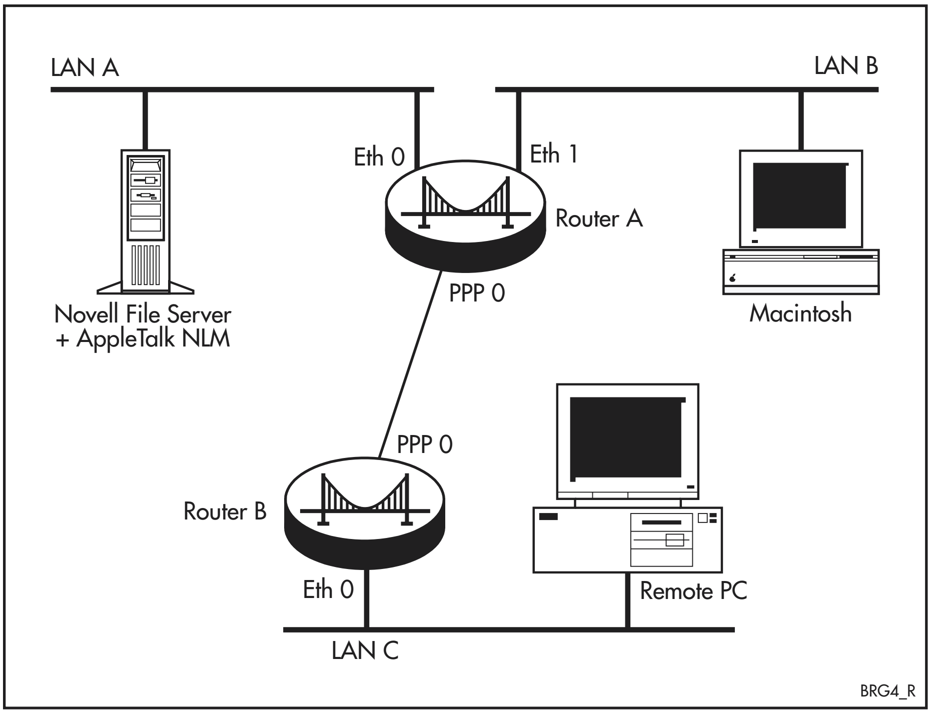
It is generally preferable to route a protocol rather than to bridge it. However, there are situations when it is more appropriate to bridge a protocol rather than route it. The Bridge module can bridge both routable and non-routable protocols. The following examples show how to configure an extended LAN.

- A Basic LAN Bridge Setup
- Bridging in a Meshed Network with Spanning Tree
- A Bridge Setup Using Filters
- VLAN-to-WAN Bridge Configuration

A Basic LAN Bridge Setup

This example shows how to configure a router to bridge AppleTalk Phase I and Novell protocols between two local Ethernet LANs and a remote LAN accessed via a wide area link (Figure 16-5 on page 16-17). The Spanning Tree Protocol is not be enabled because there is only one route between any two LANs.

Figure 16-5: Example configuration for a basic bridged network



To configure a basic LAN bridge:

1. **Configure the bridge ports.**

   Enable the bridge module using the following commands on both routers:

   ```
   create ppp=0 over=syn0
   enable bridge
   purge bridge
   enable bridge
   add bridge port=1 int=eth0
   add bridge port=2 int=eth1
   add bridge port=3 int=ppp0
   ```

   On Router B, assign the bridge ports:

   ```
   add bridge port=1 int=eth0
   add bridge port=2 int=ppp0
   ```

   The bridge module is functional after at least two ports have been assigned. This can be checked with the **show bridge port** command on page 16-63.

2. **Configure the bridge protocols.**

   Add the protocols to be bridged. Many network protocols, for example IP, have companion protocols, such as ARP, that are necessary for successful routing. When such routing protocols are bridged, their companion protocols must also be bridged. In this case, as we are bridging AppleTalk, we need to specify two protocol types, one for the AppleTalk traffic protocol and one for the AppleTalk routing protocol. Use the following commands on both routers:

   ```
   add bridge protocol=appletalk type=809B
   add bridge protocol=aarp type=80F3
   add bridge protocol=novell type=ffff
   ```

   Protocols can be checked with the **show bridge protocol** command on page 16-66.

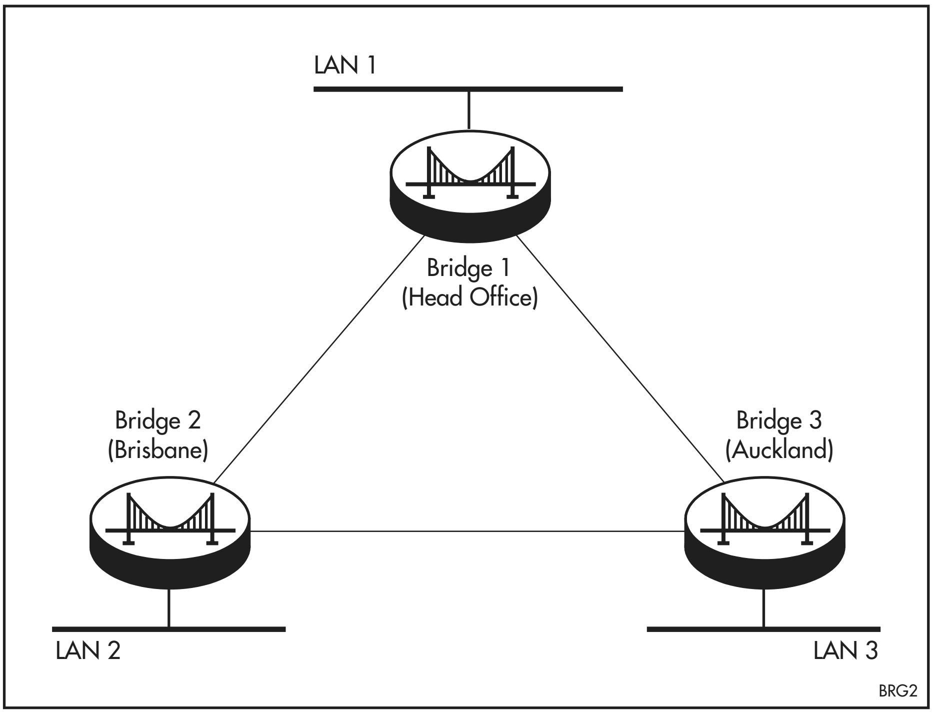
3. **Test the bridge**

   The bridge should now be tested using equipment appropriate to the protocols being bridged – in this case an AppleTalk workstation on one LAN and an AppleTalk server on the other. The amount of traffic being bridged can be monitored with the **show bridge counter** command on page 16-52.

Bridging in a Meshed Network with Spanning Tree

In this example three routers are connected in a mesh arrangement with each having a connection to each other (Figure 16-6 on page 16-19). This arrangement allows redundancy and the automatic backup of a failed router. The bridges are passing EtherTalk only.

Figure 16-6: A bridged network with a mesh topology using STP to provide redundancy



To configure bridging in a meshed network:

1. **On Bridge 1 execute the commands:**

```
enable bridge
enable bridge spanning
create ppp=0 over=syn0
create ppp=1 over=syn1
add bridge port=1 interface=eth0
add bridge port=2 interface=ppp0
add bridge port=3 interface=ppp1
add bridge group="Wide Area"
set bridge port=2 group="Wide Area"
set bridge port=3 group="Wide Area"
add bridge protocol="ethertalk arp" type="ethertalk 2
   aarp"
```

2. **On Bridge 2 execute the commands:**

```
enable bridge

enable bridge spanning

create ppp=0 over=syn0

create ppp=1 over=syn1

add bridge port=1 interface=eth0

add bridge port=2 interface=ppp0

add bridge port=3 interface=ppp1

add bridge group=“Wide Area“

set bridge port=2 group=“Wide Area“

set bridge port=3 group=“Wide Area“

add bridge protocol=“ethertalk arp“ type=“ethertalk 2
   aarp“
```

3. **On Bridge 3 execute the commands:**

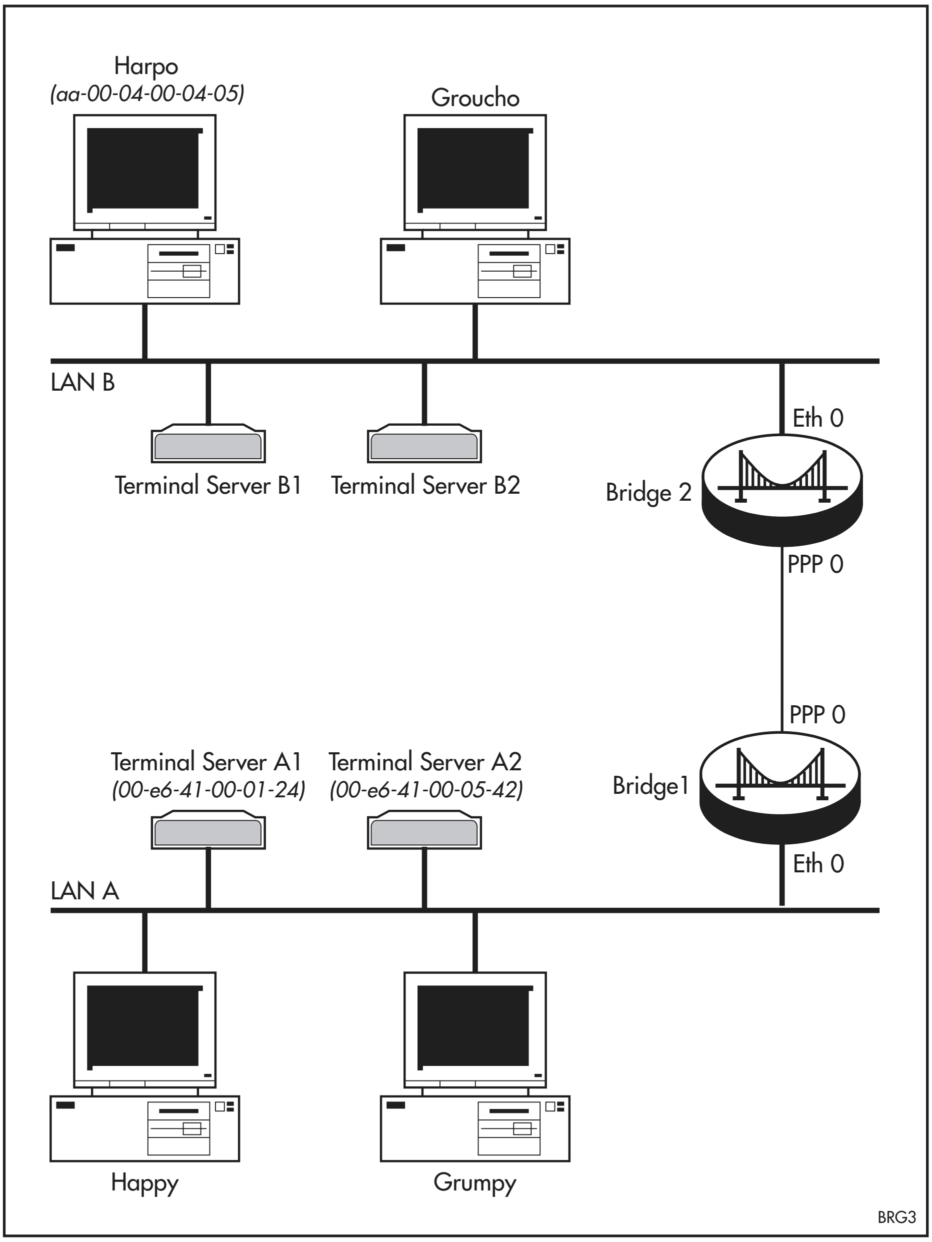
```
enable bridge

enable bridge spanning

create ppp=0 over=syn0

create ppp=1 over=syn1

add bridge port=1 interface=eth0

add bridge port=2 interface=ppp0

add bridge port=3 interface=ppp1

add bridge group=“Wide Area“

set bridge port=2 group=“Wide Area“

set bridge port=3 group=“Wide Area“

add bridge protocol=“ethertalk arp“ type=“ethertalk 2
   aarp“
```

A Bridge Setup Using Filters

In this example two LANs that historically have been separate are to be connected (Figure 16-7 on page 16-21). While other protocols are to be routed, the use of DEC LAT on each LAN in the past means that consideration must be given to bridging LAT in the new configuration.

Analysis shows that the only requirement is that terminals on LAN A be able to access the host “Harpo” on LAN B and that no other LAT traffic is required or allowed. In order to reduce the traffic and to provide some level of security for other hosts in the network, bridge filters are used to restrict the LAT traffic from LAN B to LAN A to traffic originating from host Harpo only and to restrict LAT traffic from LAN A to LAN B to traffic originating from terminal servers.

Figure 16-7: A bridged network requiring the use of bridge filters



To configure bridging with filters:

1. **On bridge 1 execute the commands:**

```
create ppp=0 over=syn0

enable bridge

add bridge port=1 int=eth0

add bridge port=2 int=ppp0

add bridge protocol=1 type="dec lat"

add bridge filter=1 sa=00-e6-41-00-01-24 port=all

add bridge filter=1 sa=00-e6-41-00-05-42 port=all

set bridge port=1 filter=1
```

2. **On bridge 2 execute the commands:**

```
create ppp=0 over=syn0

enable bridge

add bridge port=1 int=eth0

add bridge port=2 int=ppp0

add bridge protocol=1 type="dec lat"

add bridge filter=1 sa=aa-00-04-00-04-05 port=all

set bridge port=1 filter=1
```

3. **Test the configuration**

   Verify the following:

   - That the terminal servers on LAN A can see the service “Harpo”.
   - That the terminal servers on LAN A cannot see the service “Groucho”.
   - That terminals connected to the terminal servers on LAN A can connect to Harpo.
   - That the terminal servers on LAN B cannot see any of the services on LAN A.
   - That the bridge filter counters are incrementing in the expected fashion; that is, the frames matched by the filters increase steadily as terminal sessions between the terminal servers on LAN A and Harpo proceed, and that the number of frames steadily increases that are dropped as a result of not matching a filter entry.

VLAN-to-WAN Bridge Configuration

Figure 16-8 on page 16-23 shows a simple remote VLAN connection. A company has its head office at location A and its training centre at location B. In location B, a training server provides computer based training programs that are accessible from selected user PCs located at both sites. Unfortunately, the training application operates over an unroutable protocol.

To solve this problem, a single VLAN is created for the training PCs and a remote VLAN connection lets them access the wide area link.

Figure 16-8: Example configuration for a remotely bridged VLAN

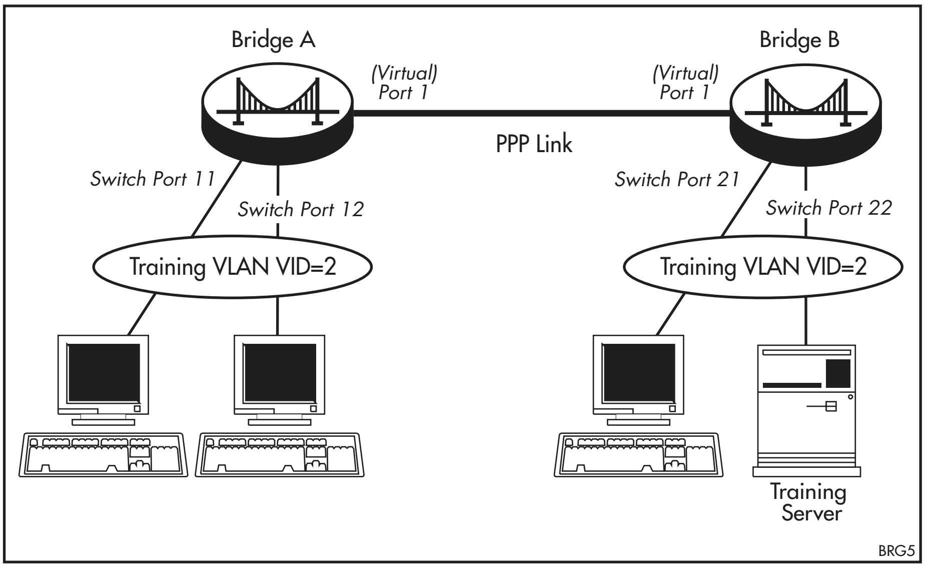


Table 16-2: VLAN membership of example of a network using tagged ports

VLAN	Member ports
Training	11, 12 on Bridge A 21, 22 on Bridge B

To configure VLAN-to-WAN bridge A

1. **Ensure that the set bridge stripvlantag command on page 16-48 is set to OFF.**

2. **Create the VLAN to be used for the training devices.**

   Because the default VLAN with VID 1 may already exist, assign a VLAN with VID 2 for the training devices with the command:

   ```
   create vlan=Training vid=2
   ```

3. **Add switch ports to the VLAN.**

   To add switch ports to the Training VLAN, use the command:

   ```
   add vlan=Training port=11,12
   ```

4. **Add the VLAN to the bridge.**

   To add the VLAN to the bridge, use the command:

   ```
   add vlan=2 bridge
   ```

5. **Create a WAN interface.**

   To create a PPP interface over a synchronous port or other interface, use the command:

   ```
   create ppp=0 over=syn0
   ```

6. **Configure the bridge ports.**

   To enable the bridge module and add the PPP interface as a virtual port, use the commands:

   ```
   enable bridge
   set bridge stripvlantag=no
   add bridge port=1 int=ppp0
   ```

To configure VLAN-to-WAN bridge B

1. **Ensure that the set bridge stripvlantag command on page 16-48 is set to OFF.**

2. **Create the Training VLAN.**

   To create a Training VLAN with VID 2 to be used for VLAN-to-WAN bridging, use the command:

   ```
   create vlan=Training vid=2
   ```

3. **Add switch ports to the VLAN.**

   To add switch ports to the Training VLAN, use the command:

   ```
   add vlan=Training port=21,22
   ```

4. **Add the VLAN to the bridge.**

   To add the VLAN to the bridge, use the command:

   ```
   add vlan=2 bridge
   ```

5. **Create a WAN interface.**

   To create a PPP interface over a synchronous port or other interface, use the command:

   ```
   create ppp=0 over=syn0
   ```

6. **Configure the bridge ports.**

   To enable the bridge module and add the PPP interface as a virtual port, use the commands:

   ```
   enable bridge
   set bridge stripvlantag=no
   add bridge port=1 int=ppp0
   ```

For more information about configuring Frame Relay, ATM, and PPP, see Chapter 14, Frame Relay, Chapter 10, ATM over xDSL, and Chapter 15, Point-to-Point Protocol (PPP).

Command Reference

This section describes the commands available on the router to enable, configure, control and monitor the bridge module.

The shortest valid command is denoted by capital letters in the Syntax section. See "Conventions" on page lxv of About this Software Reference for details of the conventions used to describe command syntax. See Appendix A, Messages for a complete list of messages and their meanings.

add bridge filter

Syntax

```
ADD BRIDge FILter=1..99 POrt={ALL|NONE|port-list}
    [ENTry=entry] [SAddress<sep1>macadd [SMask=macadd]]
    [DAddress<sep1>macadd [DMask=macadd]]
    [ENCapsulation<sep1>{802|Ethii|Snap|NOVell}
    [DIscriminator<sep1>protocoltype]] [SIze<sep2>1..65535]
    [Offset=1..1500 Data<sep1>datastring]
    [TYpe<sep1>{UNIcast|MULticast|BROadcast|ANY}]
```

where:

- *datastring* is a hex number up to 32 hex digits long that represents a sequence of bytes to match packet data. The number of hex digits must be even.
- *entry* is a filter entry number from 1 to *n*+1 where *n* is the number of filter entries currently defined in the filter.
- *macadd* is an Ethernet six-octet MAC address, expressed as six pairs of hexadecimal digits delimited by hyphens.
- *protocoltype* is either a valid protocol number or a recognised protocol name. A protocol number can be either 1 byte for SAP, 2 bytes for ETHII or 5 bytes for an 802.2 SNAP type packet, and is specified in hexadecimal.
- *sep1* is a separator, either "`=`" (is equal to) or "`!=`" (is not equal to).
- *sep2* is a separator, either "`>=`" (is greater than or equal to) or "`<=`" (is less than or equal to).
- *port-list* is a port number from 1 to 32, or a comma-separated list of port numbers.

Description This command adds a single filter entry to the bridge access filters. This entry is a condition that is imposed on all frames passing through the filter. Filter entries are applied in the order determined by the **entry** list and operate such that frames matching the selection criteria are passed to the ports defined by the **port** parameter. Filtering may be based on the following frame components: source and destination MAC addresses, frame encapsulation, protocol and discriminator size, broadcast type, and data content.

The **port** parameter specifies the ports that a frame matching this filter entry may be forwarded over. If **all** is specified, the frame is eligible for forwarding over all bridge ports. If **none** is specified, the frame may not be forwarded, and should be discarded. If a comma-separated list of ports is specified, the frame forwarding procedure decides which of the specified ports should receive the frame based upon its analysis of the bridged traffic.

The **daddress** parameter specifies the value that is matched against the destination MAC address of frames being filtered. If the **dmask** parameter is supplied, the destination MAC addresses are masked with the specified value prior to comparison with **daddress**. The default is to match any destination MAC address.

The **data** parameter specifies the data to match, starting at the offset given by the **offset** parameter. Up to 16 bytes of data can be matched, to either check that the data is present (=) or is not present (!=). If the **data** parameter is specified, the **offset** parameter must also be specified.

The **discriminator** parameter specifies a value to match in the protocol field of the frame. For Ethernet-II frames, an 8-bit value (two hexadecimal digits) is required. For 802.2 frames, a 16-bit value (four hexadecimal digits) is required. For SNAP frames a 5-byte value (ten hexadecimal digits) is required. Optionally, a keyword like the keywords used in the **add bridge protocol** command on page 16-30 may be entered, except that the keywords “ALL802”, “ALLETHII”, “ALLSNAP” and “NOVELL” are not allowed. If **discriminator** is specified, the **encapsulation** parameter must also be present and specify an encapsulation other than NOVELL, and the separator used with the **encapsulation** parameter must be “=”.

The **dmask** parameter specifies a (bitwise) mask to apply to destination MAC addresses from frames prior to comparison with the **daddress** value. If **dmask** is specified, **daddress** must also be specified. The default is **ff-ff-ff-ff-ff-ff**.

The **encapsulation** parameter specifies the format of the frames that match this filter entry. The four possible settings correspond to the frame types supported by the bridge module—Ethernet-II, IEEE 802.2, SNAP and Novell’s 802.3 format. The default is to match any type of frame. This parameter must be specified if the **discriminator** parameter is used.

The **entry** parameter specifies where in a filter list the new entry is added. If **entry** is not specified the new entry is added to the end of the filter list. If specified, its value cannot be greater than one more than the number of entries in the list.

The **filter** parameter specifies the filter to which the entry is added. If the filter does not exist, it is created. In this case, **entry** must either be unspecified or be set to 1.

The **offset** parameter indicates an offset in the Ethernet packet being checked for filtering, starting at the first octet in the user data part of the packet. Source and destination address, layer 2 fields (including VLAN tag), protocol type fields, and CRC are not part of the user data. The first octet in the user data is at offset 1 for the purposes of data filtering. The **offset** parameter must be specified if the **data** parameter is specified, and is invalid otherwise.

The **saddress** parameter specifies the value to match against the source MAC address in a frame. If the **smask** parameter is supplied, it is used to bitwise-AND the source MAC address from the frame prior to comparison with **saddress**. The default is to match any source MAC address.

The **size** parameter specifies the size of the user data part of the frame matching this filter entry. Source and destination address, layer 2 fields (including VLAN tag), protocol type fields and CRC are not part of the user data. The size of the frame is taken by excluding the address, type/length field and protocol discriminator. Any value from 1 to 65535 may be entered, but only a subset of this range is sensible in most networks. For example, the size of an

Ethernet frame is between 64 and 1514 bytes. The separator for this parameter must be either "<=" or ">=", which means that the filter entry always matches a range of frame sizes. The default is to match any frame size.

The **smask** parameter specifies a (bitwise) mask to apply to source MAC addresses from frames prior to comparison with the **saddress** value. If **smask** is specified, **saddress** must also be specified. The default is **ff-ff-ff-ff-ff-ff**.

The **type** parameter specifies the broadcast/multicast type to match. If **broadcast** is specified, the filter matches broadcast frames with destination MAC address **ff-ff-ff-ff-ff-ff**. If **multicast** is specified, the filter matches all non-unicast frames with the multicast bit set in the first octet of the MAC address (including broadcast frames). If **unicast** is specified, the filter matches frames directed to a particular station. The default is to match any type.

Examples

To add a filter entry to bridge filter number 1 that rejects any 802.2-framed IP ARP frames from station 00-00-cd-12-34-56, use the command:

```
add brid fil=1 sa=00-00-CD-12-34-56 enc=802 di=0806 po=none
```

To add an entry to filter 1 that rejects frames that have a destination address of **`ff-ff-ff-ff-ff-ff`**, Novell encapsulation, and in which the byte at offset 47 in the data field is not 41, use the command:

```
add brid fil=1 da=ff-ff-ff-ff-ff-ff enc=novell o=47
    data!=41 po=none
```

Where a filter is applied, frames are discarded that do not meet the criteria in at least one filter entry. A general filter entry can be added to the end of a new filter to ensure that frames not explicitly filtered out are passed on by the filter, if this is required. Use the command **add bridge filter=1 port=all**.

Related Commands

delete bridge filter
set bridge filter
show bridge filter

add bridge group

Syntax

```
ADD BRIDge GROup={groupname|1..32} [REclustering=0..15]
```

where *groupname* is a unique name for the group up to 32 characters long

Description

This command adds a group to the Spanning Tree Protocol. The command specifies the group’s number and optionally associates a name, up to 32 characters long, with the group. The group name should be unique.

The **reclustering** parameter sets the upper bound for the expected time that it takes for new cluster membership information to propagate throughout the group. The default is 4 seconds.

The reclustering delay should be set to zero for a virtual LAN (a group to which each member remote bridge attaches by a single virtual port).

Groups are numbered consecutively starting at group number one. The group number is used only on the local bridge.

Examples

To add a group called “Western” to the Spanning Tree Protocol, use the command:

```
add brid gro=“Western”
```

Related Commands

delete bridge group
set bridge group
show bridge group

add bridge port

Syntax

```
ADD BRIDge POrt=1..32 INTerface=interface
    [CIRCuit=circuit]
```

where:

- *interface* is a valid interface name.
- *circuit* is a circuit number within an interface that supports multiple logical connections per physical connection. For a Frame Relay interface, *circuit* maps to a DLCI. Bridging is not supported over X.25 interfaces.

Description

This command adds an interface for use by the bridge module. The command must be executed for each interface that is to be used by the bridge module. The bridge module does not become active until at least two ports have been defined. Each port on a router must be given a unique number from 1 to 32.

This command adds an interface as a virtual port on the bridge module. The command must be executed for each interface that is to be added. The bridge module is not active until at least two ports have been added, or a single virtual port has been added and the bridge has also been added to a VLAN.

The **port** parameter specifies a unique virtual port to be added to the bridge.

When operating the bridge in the VLAN-to-WAN mode, it can have one virtual port. If there are already two or more virtual ports on the bridge, the **delete bridge port** command should be used to reduce the number of virtual ports to one.

The **interface** parameter specifies the router interface to be assigned to the logical bridge port. The **interface** parameter specifies the interface to be added to a virtual port. Valid interfaces are:

- eth (such as eth0)
- ATM (such as atm0.1)
- PPP (such as ppp0)
- FR (such as fr0)

The interface must already exist. To see a list of all currently available interfaces, use the command **show interface**.

The **circuit** parameter is required for Frame Relay interfaces but is invalid for other interface types.

Example

To add PPP interface 0 as bridge port 1, use the command:

```
add brid po=1 int=ppp0
```

To add Ethernet interface 0 as bridge port 1, use the command:

```
add brid po=1 int=eth0
```

Related Commands

delete bridge port
set bridge port
show bridge port

add bridge protocol

Syntax

```
ADD BRIDge PROTocol[=label|index] {TYpe=protocolname|
   protocolnumber} [PRIOrity=priority]
```

where:

- *label* is a user-defined descriptive name for the protocol entry, up to 32 characters long.
- *index* is a user-defined index for the protocol entry.
- *protocolname* is a valid protocol name as listed in Table 16-3.
- *protocolnumber* is a valid protocol number as listed in Table 16-3. A protocol number can be either 1 byte for SAP, 2 bytes for ETHII or 5 bytes for an 802.2 SNAP type packet, and is specified in hexadecimal.
- *priority* is a forwarding priority from 0 (lowest) to 4 (highest).

Description

This command adds a type of protocol to be bridged. Except when using the VLAN-WAN bridging, only protocols that have been specifically enabled are bridged. This conservative approach limits the traffic being transported across potentially slower WAN links when the bridge module is activated.

The bridge module provides a predefined list of common protocols. To bridge one of these, specify the protocol name (from Table 16-3 on page 16-30) as the value for the **type** parameter and the bridge module automatically substitutes the correct protocol number. The bridge also has the option of bridging all protocols of a given encapsulation type by using the keywords ALL802, ALLETHII, and ALLSNAP. The protocol name All802 selects all 802.2 encapsulation packets, including SNAP and Novell802.3 encapsulation packets.

A number of network protocols (e.g. IP, AppleTalk, EtherTalk) have companion protocols (for example RIP, AppleTalk AARP, EtherTalk AARP) that are necessary for successful routing. When these network protocols are bridged, their companion protocols must also be bridged. For example, to bridge AppleTalk Phase I, AppleTalk (0x809B) and AppleTalk AARP (0x80F3) must also be bridged. To bridge AppleTalk Phase II (EtherTalk II), EtherTalk 2 (0x080007809B) and EtherTalk 2 AARP (0x00000080F3) must also be bridged.

Note that you cannot add the PPP protocol to a port that is also a PPPoE end point. **create ppp** command on page 15-56 of Chapter 15, Point-to-Point Protocol (PPP).

Table 16-3: Predefined protocol types implemented by the bridge module

Protocol Name	Protocol Number	Encapsulation
All802	all 802.2 protocols	SAP
EIA-RS	004E	SAP
ISO CLNS IS	00FE	SAP
Netbeui	00F0	SAP
Novell 802.2	00E0	SAP
PROWAY	008E	SAP
PROWAY-LAN	000E	SAP

Table 16-3: Predefined protocol types implemented by the bridge module (cont.)

Protocol Name	**Protocol Number**	**Encapsulation**
SNA Path Control	0004	SAP
Appletalk	809B	EthII
AppleTalk AARP	80F3	EthII
ARP	0806	EthII
Banyan Systems	0BAD	EthII
BBN Simnet	5208	EthII
Chaosnet	0804	EthII
DEC Customer	6006	EthII
DEC DECNET	6003	EthII
DEC Diagnostic	6005	EthII
DEC Encryption	803D	EthII
DEC LANBridge	8038	EthII
DEC LAT	6004	EthII
DEC LAVC	6007	EthII
DEC MOP Dump/Ld	6001	EthII
DEC MOP Rem Cons	6002	EthII
ECMA Internet	0803	EthII
IBM SNA	80D5	EthII
IP	0800	EthII
NBS Internet	0802	EthII
Novell II	8137	EthII
PPPoE Discovery	8863	EthII
PPPoE Session	8864	EthII
PUP Addr Trans	0A01	EthII
RARP	8035	EthII
SNMP	814C	EthII
X.25 Level 3	0805	EthII
X.75 Internet	0801	EthII
XEROX NS IDP	0600	EthII
XEROX PUP	0A00	EthII
XNS Compat	0807	EthII
Novell802.3	FFFF	Novell
AllSNAP	all SNAP protocols	SNAP
ETHERTALK 2	080007809B	SNAP
ETHERTALK 2 AARP	00000080F3	SNAP
Novell SNAP	0000008137	SNAP

Note: When you enter a protocol name that contains spaces, you must surround the name with double quotation marks. You can use lowercase or uppercase letters. For example, to specify ETHERTALK 2 AARP, enter **type=”ethertalk 2 aarp”**.

The **priority** parameter specifies the priority for forwarding frames of this protocol over a wide area link. Frames with a higher priority are forwarded before frames with a lower priority. If a frame matches two or more bridged protocols of the same encapsulation, each with priorities set, the most specific protocol is used to set the frame’s priority. The default priority is 1.

Examples

To bridge LAT frames, use either of the following commands:

```
add brid prot="DIGITAL LAT" ty=6004
```

```
add brid prot="DIGITAL LAT" ty="DEC LAT"
```

To bridge IP packets at priority 2 and all other protocols with Ethernet II encapsulation at priority 1, use the commands:

```
add brid prot=1 ty=ALLETHII prio=1
```

```
add brid prot=1 ty=ip prio=2
```

Related Commands

delete bridge protocol
set bridge protocol
add bridge port
delete bridge port
show bridge protocol

add bridge station

Syntax

```
ADD BRIDge STation=macadd POrt=1..32
```

where *macadd* is an Ethernet six-octet MAC address, expressed as six pairs of hexadecimal digits delimited by hyphens

Description

This command adds a single entry to the bridge station map. The bridge station map is a list of MAC addresses known to the bridge and associated with the port where the station can be found. Normally, bridge station map entries are learned by inspecting the frames that the bridge receives, but this command exists to add static entries to the bridge station map. Note that bridge station map is not active while the bridge is operating in the VLAN-to-WAN mode.

The entries added to the bridge station map by this command become part of the bridge module configuration, so can be saved with the **create config** command on page 5-22 of Chapter 5, Managing Configuration Files and Software Versions.

The **station** parameter specifies the MAC address of the station being added to the bridge station map.

The **port** parameter specifies the port out which the station is found. The specifiedport must exist.

Examples

To add a bridge station map entry for MAC address 00-00-cd-12-34-56, which is reached via port 1, use the command:

```
add brid st=00-00-cd-12-34-56 po=1
```

Related Commands

delete bridge station
show bridge station

add vlan bridge

Syntax

```
ADD VLAN={vlan-name|1..4094} BRIDge
    [DEVICELimit={NONE|1..250}]
    [AGEingtimer={NONE|0..10000001}]
```

- where *vlan-name* is a unique name for the VLAN 1 to 32 characters long. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**. Up to 16 VLANs can be configured using this command, but each must be separately entered.

Description

This command enables bridging between the switch ports that are members of the specified VLAN, and a single virtual port configured on the bridge. The VLAN forwards all frames to the bridge’s single virtual port. Frames destined for remote stations are forwarded to the wide area port. Frames destined for stations on the local bridge are sent to the VLAN and port appropriate to that station.

The **devicelimit** parameter sets the maximum number of devices connected to the VLAN that can send packets over the VLAN to WAN bridge. Specify **none** to set no limit to the number of devices. Default: **none** (no limit).

The **ageingtimer** parameter sets the number of seconds before an unused MAC entry is removed. Specify **none** to set no time limit. Default: **none** (no time limit).

For PPP operation a maximum of 16 VLANs can be attached to the bridge. When multiple VLANs are attached to the bridge, all the frames transmitted or received by the bridge must be VLAN tagged. To ensure this you must configure the **set bridge stripvlantag** to **no**.

Examples

To attach the training VLAN to the bridge, use the command:

```
add vlan=training bridg
```

To attach the training VLAN to the bridge for 20 devices with an ageing timer of 1 hour (3600 seconds), use the command:

```
add vlan=training bridg devicel=20 age=3600
```

Related Commands

add bridge port
delete vlan bridge
enable bridge
set bridge stripvlantag
show bridge
show vlan in Chapter 8, Switching

delete bridge filter

Syntax

```
DELete BRIDge FILter=1..99 [ENTry=entry]
```

where *entry* is a filter entry number from 1 to *n*+1 where *n* is the number of filter entries currently defined in the filter

Description

This command deletes a single bridge filter entry, or an entire filter. Bridge filtering is inactive in Remote VLAN mode.

The **filter** parameter specifies the bridge filter containing the filter entry to be deleted. The filter must exist.

The **entry** parameter specifies the particular filter entry within the selected filter to be deleted. The filter entry must exist. If a filter entry is not specified, the entire filter is deleted.

Examples

To delete filter entry 2 within bridge filter 3, use the command:

```
del brid fil=3 ent=2
```

Related Commands

add bridge filter
set bridge filter
show bridge filter

delete bridge group

Syntax

```
DELete BRIDge GROup{=groupname|1..32}
```

where *groupname* is a unique name for the group, up to 32 characters long

Description

This command deletes a group from the Spanning Tree Protocol. Either the group number or the group name must be specified. The group must have been defined previously with the **add bridge group** command on page 16-28.

Examples

To delete group “Western” from the Spanning Tree Protocol, use the command:

```
del brid gro=“Western“
```

Related Commands

add bridge group
set bridge group
show bridge group

delete bridge port

Syntax `DELete BRIDge POrt=1..32`

Description This command removes a port from use by the bridge module.

Examples To delete bridge port 2, use the command:

```
del brid po=2
```

Related Commands **add bridge port**
set bridge port
show bridge port

delete bridge protocol

Syntax `DELete BRIDge PROTocol{=`*`protocolname`*`|`*`index`*`}`

where:

- *protocolname* is a descriptive name for the protocol entry that was assigned at the time the protocol was added, up to 32 characters long.
- *index* is a manager-defined index for the protocol entry.

Description This command deletes a protocol from the Bridge module so that it is no longer bridged. The protocol to be deleted can be specified by the manager-assigned protocol name or by index, which can be found in the output of the **show bridge protocol** command on page 16-66.

Examples To delete the bridge protocol with an index of 4, use the command:

```
del brid prot=4
```

Related Commands **add bridge protocol**
show bridge protocol

delete bridge station

Syntax

```
DELete BRIDge STation=macadd POrt=1..32
```

where *macadd* is an Ethernet six-octet MAC address, expressed as six pairs of hexadecimal digits delimited by hyphens

Description

This command deletes a single entry from the bridge station map. This is a list of MAC addresses known to the bridge and associated with the port where the station can be found. This command deletes one of these entries, including entries that have been learned in the filtering and forwarding process, but not addresses of type “self”.

The **station** parameter specifies the MAC address of the station being deleted from the bridge station map.

The **port** parameter specifies the port over which the station is found. This must be specified in order to locate the bridge station map entry correctly.

Examples

To delete the bridge station map entry for MAC address 00-00-cd-12-34-56, which is out port 1, use the command:

```
del brid st=00-00-cd-12-34-56 po=1
```

Related Commands

add bridge station
show bridge station

delete vlan bridge

Syntax

```
DElete VLAN={vlan-name|1..4094} BRIDge
```

where *vlan-name* is a unique name for the VLAN 1 to 32 characters long. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.

Description This command deletes a bridge attachment from the specified VLAN.

Examples To delete the training VLAN from the bridge use the command:

```
del vlan=training brid
```

Related Commands

add vlan bridge
show bridge

disable bridge

Syntax

```
DISable BRIDge
```

Description This command disables the bridge module and takes effect immediately.

Examples To disable bridging, use the command:

```
dis brid
```

Related Commands

enable bridge
purge bridge
reset bridge

disable bridge learning

Syntax

```
DISable BRIDge LEarning
```

Description This command disables the dynamic learning and updating of the bridge source bridge station map.

If learning is disabled, statically entered MAC source addresses or protocol types are used to decide which packets to forward. All existing dynamically learned filters age and are lost.

Examples To disable the bridge learning function, use the command:

```
dis brid le
```

Related Commands

enable bridge learning
show bridge

disable bridge spanning

Syntax

```
DISable BRIDge SPanning
```

Description

This command disables the Spanning Tree Protocol, and resets the Bridge module. The Spanning Tree Protocol should be switched off if the network topology is free of loops. If a loop exists, the performance loss may be considerable. The Spanning Tree Protocol is disabled by default.

Examples

To disable the Spanning Tree Protocol, use the command:

```
dis brid sp
```

Related Commands

enable bridge spanning
set bridge spanning
show bridge
show bridge spanning

enable bridge

Syntax

```
ENAble BRIDge
```

Description

This command enables the bridge module and takes effect immediately. The bridge module must be properly configured and enabled; if it has not previously been configured, then all (requisite) parameters are set to their defaults.

Examples

To enable bridging, use the command:

```
ena brid
```

Related Commands

disable bridge
purge bridge
reset bridge

enable bridge learning

Syntax

```
ENAble BRIDge LEarning
```

Description

This command enables the dynamic learning and updating of the bridge station map.

Examples

To enable the bridge learning function, use the command:

```
ena brid le
```

Related Commands

disable bridge learning
show bridge

enable bridge spanning

Syntax

```
ENAble BRIDge SPanning
```

Description

This command enables the use of the Spanning Tree Protocol. The Spanning Tree Protocol should be switched off if the network topology is free of loops. If a loop exists in the topology, then the performance loss may be considerable. This command results in a reset of the bridge module. The Spanning Tree Protocol is disabled by default.

STP can be used to prevent loops between fixed ports and ports in expansion options, but does not prevent loops within a VLAN. The network administrator must ensure that there are no physical loops within a VLAN.

Examples

To enable the Spanning Tree protocol, use the command:

```
ena brid sp
```

Related Commands

disable bridge spanning
set bridge spanning
show bridge
show bridge spanning

purge bridge

Syntax

```
PURge BRIDge
```

Description

This command removes bridge configuration information, resets all bridge counters, and restores all defaults. It destroys the association between all existing VLANs and the bridge. The command should be used before making major changes to the configuration data.

Examples

To purge the current bridge configuration, use the command:

```
pur brid
```

Related Commands

disable bridge
enable bridge
reset bridge

reset bridge

Syntax

```
RESET BRIDge
```

Description

This command resets the bridge module. The dynamic filtering database is cleared and initialized with entries from the permanent filtering database, and the bridge protocol entity is initialised.

Examples

To reset the bridge module, use the command:

```
reset brid
```

Related Commands

disable bridge
enable bridge
purge bridge

set bridge ageingtimer

Syntax

```
SET BRIDge AGEingtimer=10..1000000
```

Description

This command sets the threshold value, in seconds, of the ageing timer, after which a dynamic entry in the filtering database is automatically removed. The default is 300 seconds.

Examples

To set the ageing timer to 180 seconds, use the command:

```
set brid age=180
```

Related Commands

show bridge
show bridge filter

set bridge filter

Syntax

```
SET BRIDge FILter=1..99 ENTry=entry [SAddress<sep1>macadd
    [SMask=macadd]] [DAddress<sep1>macadd [DMask=macadd]]
    [ENCapsulation<sep1>{802|Ethii|Snap|NOVell}
    [DIscriminator<sep1>protocoltype]] [SIze<sep2>1..65535]
    [Offset=1..1500 Data<sep1>datastring]
    [TYpe<sep1>{UNIcast|MULticast|BROadcast|ANY}]
    POrt={ALL|NONE|1..32[,1..32]...}
```

where:

- *entry* is a filter entry number from 1 to *n* where *n* is the number of filter entries currently defined in the filter.
- *macadd* is an Ethernet six-octet MAC address, expressed as six pairs of hexadecimal digits delimited by hyphens.
- *protocoltype* is either a valid protocol number or a recognised protocol name. A protocol number can be either 1 byte for SAP, 2 bytes for ETHII or 5 bytes for an 802.2 SNAP type packet, and is specified in hexadecimal.
- *sep1* is a separator, either "`=`" (is equal to) or "`!=`" (is not equal to).
- *sep2* is a separator, either "`>=`" (is greater than or equal to) or "`<=`" (is less than or equal to).

Description This command modifies the settings of a single bridge access filter entry. This entry is a condition imposed on frames passing through the filter. Filter entries are applied in the order determined by an entry list and operate such that frames matching the selection criteria are passed to the ports defined by the **port** parameter. Filtering may be based on the following frame components: source and destination MAC addresses, frame encapsulation, size, broadcast type, protocol discriminator, and contents of data.

This command does not allow options that were previously on to be turned off. For example, if a filter entry on source address was created with a command like **add bridge filter=1 SA=00-00-cd-00-00-00 sm=ff-ff-ff-00-00-00**, you cannot set the filter so that it does not filter on source address with the **set bridge filter** command. In this case, the filter entry should be deleted and a new one created.

The **entry** parameter specifies an existing filter entry to be modified. Where a filter is applied, frames are discarded that do not meet the criteria in at least one filter entry.

The **daddress** parameter specifies the value to match against the destination MAC address from frames being filtered. If the **dmask** parameter is supplied, the destination MAC addresses are masked with the specified value prior to comparison with **daddress**. The default is to match any destination MAC address.

The **data** parameter specifies the data to match, starting at the offset given by the offset parameter. Up to 16 bytes of data can be matched, to either check that the data is present (=) or is not present (!=). If the **data** parameter is specified, the **offset** parameter must also be specified.

The **dmask** parameter specifies a (bitwise) mask to apply to destination MAC addresses from frames prior to comparison with the **daddress** value. If **dmask** is specified, **daddress** must also be specified. The default is **ff-ff-ff-ff-ff-ff**.

The **discriminator** parameter specifies a value to match in the protocol field of the frame. For Ethernet-II frames, an 8-bit value (two hexadecimal digits) is required. For 802.2 frames, a 16-bit value (four hexadecimal digits) is required. A 5-byte value (ten hexadecimal digits) is required for SNAP frames. Optionally, a keyword like the keywords used in the **add bridge protocol** command on page 16-30 may be entered, except that the keywords ALL802, ALLETHII, ALLSNAP, and NOVELL are not allowed. If **discriminator** is specified, the **encapsulation** parameter must also be present and specify an encapsulation other than **novell**, and the separator used with the **encapsulation** parameter must be "=".

The **encapsulation** parameter specifies the format of the frames that should match this filter entry. The four possible settings correspond to the frame types supported by the bridge module: Ethernet-II, IEEE 802.2, SNAP, and Novell's 802.3 format. The default is to match any type of frame. This parameter must be specified if the **discriminator** parameter is used.

The **filter** parameter identifies the filter containing the entry to be modified. The specified filter must exist.

The **offset** parameter indicates an offset in the Ethernet packet being checked for filtering, starting at the first octet in the user data part of the packet. Source and destination address, layer 2 fields (including VLAN tag), protocol type fields, and CRC are not part of the user data. The first octet in the user data is at offset 1 for the purposes of data filtering. The **offset** parameter must be specified if the **data** parameter is specified, and is invalid otherwise.

The **port** parameter specifies the ports that a frame matching this filter entry may be forwarded over. If **all** is specified, the frame is eligible for forwarding over all bridge ports. If **none** is specified, the frame may not be forwarded, and should be discarded. If a comma-separated list of ports is specified, the frame forwarding procedure decides which of the specified ports should receive the frame based upon its analysis of the bridged traffic.

The **saddress** parameter specifies the value to match against the source MAC address in a frame. If the **smask** parameter is supplied, it is used to bitwise-AND the source MAC address from the frame prior to comparison with **saddress**. The default is to match any source MAC address.

The **size** parameter specifies the size of the user data part of the frame matching this filter entry. Source and destination address, layer 2 fields (including VLAN tag), protocol type fields and CRC are not part of the user data. The size of the frame is taken by excluding the address, type/length field and protocol discriminator. Any value from 1 to 65535 may be entered, but a subset of this range is sensible in most networks. For example, the size of an Ethernet frame is between 64 and 1514 bytes. The separator for this parameter must be either "<=" or ">=", which means that the filter entry always matches a range of frame sizes. The default is to match any frame size.

The **smask** parameter specifies a (bitwise) mask to apply to source MAC addresses from frames prior to comparison with the **saddress** value. If **smask** is specified, **saddress** must also be specified. The default is **ff-ff-ff-ff-ff-ff.**

The **type** parameter specifies the broadcast/multicast type to match. If **broadcast** is specified, the filter matches broadcast frames with destination MAC address **ff-ff-ff-ff-ff-ff**. If **multicast** is specified, the filter matches all non-unicast frames with the multicast bit set in the first octet of the MAC address (including broadcast frames). If **unicast** is specified, the filter matches frames directed to a particular station. The default is to match any type.

Examples To modify filter entry 4 on filter 2 to apply to SNAP format frames, use the command:

```
set brid fil=2 ent=4 enc=s
```

Related Commands

add bridge filter
delete bridge filter
show bridge filter

set bridge group

Syntax

```
SET BRIDge GROup={groupname|1..32} REclustering=0..15
```

where *groupname* is a unique name for the group, up to 32 characters long

Description This command alters the reclustering parameter for an existing group. The group may be specified by either the group number or name.

The **reclustering** parameter sets the upper bound for the expected time for new cluster membership information to propagate throughout the group. The default is 4 seconds.

The reclustering delay should be set to zero for a virtual LAN (a group to which each member remote bridge attaches by a single virtual port).

Examples To set the expected time for propagation of new cluster membership information through group “Western” to the maximum value, use the command:

```
set brid gro=“Western“ re=15
```

Related Commands

add bridge group
delete bridge group
show bridge group

set bridge port

Syntax

```
SET BRIDge POrt=1..32 ACcesspriority=0..7]
    [PAthcost=1..1000000] [PRIOrity=0..255]
    [STAte={DIsabled|BLocking}] [GROup={groupname|1..32}]
    [USerpriority=0..7]
```

where *groupname* is a unique name for the filter, up to 32 characters long

Description

This command sets or changes the parameter values for a specified port.

The **userpriority** parameter controls the outbound user priority for relayed frames. The default is dependent on the MAC type but is typically 0.

The **accesspriority** parameter sets the outbound access priority. The default is 0.

The **priority** parameter sets the value of the priority field contained in the port identifier. The Spanning Tree Protocol uses the port priority when determining the root port for each bridge. The port with the lowest value is considered to be at the highest priority. The default is 128.

The **pathcost** parameter sets the path cost for each port. The **pathcost** for a LAN port should be set to a maximum of 65535. If the port is to be the root port then this value determines the total cost from the bridge to the root. The pathcost can be calculated using the following formula:

```
PATHCOST = 1000/Port_Speed_in_MB_per_second
```

The default for **pathcost** is 100 irrespective of the interface type and speed.

If the **state** parameter is set to **disabled**, the port does not participate in frame bridging or in the operation of the Spanning Tree Algorithm and Protocol. Setting the **state** parameter to **blocking** initiates the enable port procedure. During this blocking period, all frame bridging on the port ceases.

The **group** parameter specifies the group to which the port belongs. This is required for virtual ports.

Examples

To add a filter to bridge port 2, use the command:

```
set bridge po=2
```

Related Commands

add bridge port
delete bridge port
show bridge port

set bridge protocol

Syntax

```
SET BRIDge PROTocol[=protocolname|index] [PRIOrity=0..4]
```

where:

- *protocolname* is a descriptive protocol name up to 32 characters long.
- *index* is a manager-definable index for the protocol entry.

Description

This command modifies the priority of a protocol to be bridged.

The bridge module provides a predefined list of common protocols. To bridge either of these, specify the protocol name (from Table 16-3 on page 16-30) as the value for the **type** parameter, and the bridge module automatically substitutes the correct protocol number. The bridge also has the option of bridging all protocols of a given encapsulation type by using the keywords ALL802, ALLETHII, and ALLSNAP.

A number of network protocols such as IP, AppleTalk, and EtherTalk have companion protocols, for example RIP, AppleTalk AARP, and EtherTalk AARP that are necessary for successful routing. When these network protocols are bridged, their companion protocols must also be bridged. For example, to bridge AppleTalk Phase I, AppleTalk (0x809B) and AppleTalk AARP (0x80F3) must also be bridged. To bridge AppleTalk Phase II (EtherTalk II), EtherTalk 2 (0x080007809B) and EtherTalk 2 AARP (0x00000080F3) must also be bridged.

The **priority** parameter specifies the priority for forwarding frames of this protocol over a wide area link. Frames with a higher priority are forwarded first. If a frame matches two or more bridged protocols of the same encapsulation, each with priorities set, the most specific protocol is used to set the frame's priority. The default priority is 1.

Examples

To change the priority of all protocols with Ethernet II encapsulation to 3, use the commands:

```
set brid prot=1 ty=allethii prio=3
```

Related Commands

delete bridge protocol
set bridge protocol
show bridge protocol

set bridge spanning

Syntax

```
SET BRIDge SPanning [PRIOrity=0..65535] [MAxage=6..40]
    [HEllotime=1..10] [FOrwarddelay=4..30] [DEBug={ON|OFF}]
```

Description

This command sets the parameters used to control the Spanning Tree Protocol.

The Spanning Tree Protocol uses the value of the bridge priority when selecting the root bridge. The lower the value of the bridge identifier the higher the priority. The value of the **priority** parameter is used to set the writable portion of the bridge ID, i.e. the first two octets of the (8-octet long) Bridge Identifier. The remaining 6 octets of the bridge ID are given by a MAC address of one of the interfaces. The Bridge Identifier parameter is used in all configuration Spanning Tree Protocol packets transmitted by the bridge. The default for **priority** is 32768.

The **maxage** parameter sets the maximum age, in seconds, of Spanning Tree Protocol information learned over the network on any port before it is discarded. The default is 20.

The **hellotime** parameter sets the amount of time, in seconds, between the transmission of bridge Spanning Tree configuration information when it is the root of the spanning tree or is trying to become so. The default is 2 seconds.

The **forwarddelay** parameter sets the time, in seconds, used to control how fast a port changes its spanning state when moving towards the Forwarding state. The value determines how long the port stays in each of the Listening and Learning states that precede the Forwarding state. This value is used when the bridge is acting as the root. Any bridge not acting as the root bridge uses a dynamic value for **forwarddelay** set by the root bridge. The default for **forwarddelay** is 15 seconds.

The **debug** parameter enables the sending of the Spanning Tree Protocol debugging log messages and optional SNMP traps. The default is **off**.

Examples

To enable debugging of the Spanning Tree Protocol, use the command:

```
set brid sp deb=on
```

Related Commands

disable bridge spanning
enable bridge spanning
show bridge spanning

set bridge stripvlantag

Syntax

```
SET BRIDge STripvlantag={ON|OFF|YES|NO|True|False}
```

Description

This command determines whether the bridge strips out the VLAN tag of tagged packets, when it receives them on Eth or VLAN interfaces and bridges them.

If you specify the parameters **on**, **yes**, or **true**, the bridge strips the tag.
If you specify **off**, **no**, or **false**, the bridge retains the tag.
The default is **on**.

Examples

To retain the VLAN tag, use the command:

```
set brid st=off
```

Related Commands

show bridge

show bridge

Syntax `SHow BRIDge`

Description This command displays configuration information for the bridge (Figure 16-9, Table 16-4 on page 16-50).

Figure 16-9: Example output from the **show bridge** command with no VLANs attached

```
Remote Bridge
----------------------------------------------------------
Bridge Address          : 00-00-cd-00-0d-4d
Bridge Name             : Router software version 7.4
Spanning Tree Protocol  : ON
Filter Learning         : ON
Number LAN Ports        : 2
   Port Number          : 1
   Port Address         : 00-00-cd-00-0d-4d
   CAM                  : Enabled
   Port Number          : 2
   Port Address         : 00-00-cd-00-0d-82
   CAM                  : Enabled
Number Virtual Ports    : 1
   Port Number          : 3
Number of Groups        : 1
Ageingtime              : 300
Uptime                  : 12133
----------------------------------------------------------
```

Figure 16-10: Example output from the **show bridge** command with the VLAN *remote office* attached

```
Remote Bridge
-------------------------------------------------------------
Bridge Address           : 00-00-cd-00-0d-4d
Bridge Name              : Router software version 2.4.1
Address Learning         : ON
Bridge Tagging (Local)   : ENABLED
Bridge Tagging (Peer)    : ENABLED
Number LAN Ports         : 2
VLAN Attached            : remote office
Number of virtual Ports  : 1
   Port Number(s)        : 3
Ageingtime               : 300
Uptime                   : 12133
StripVlantag             : FALSE
checkVlantag             : FALSE
                          (overridden TRUE for multi VLAN)
-------------------------------------------------------------
```

Figure 16-11: Example output from the **show bridge** command with 2 VLANs attached

```
Remote Bridge
-------------------------------------------------------------
Bridge Address          : 00-00-cd-00-0d-4d
Bridge Name             : Router software version 2.4.1
Spanning Tree Protocol  : ON
Address Learning        : ON
Filter Learning         : ON
Bridge Tagging (local)  : Disabled
Bridge Tagging (Peer)   : Disabled
VLAN Attached           : 2
Number LAN Ports        : 2
   Port Number          : 1
   Port Address         : 00-00-cd-00-0d-4d
   CAM                  : Enabled
   Port Number          : 2
   Port Address         : 00-00-cd-00-0d-82
   CAM                  : Enabled
Number Virtual Ports    : 1
   Port Number          : 3
Number of Groups        : 1
Ageingtime              : 300
Uptime                 : 12133
checkVlantag           : FALSE (overridden TRUE for multi VLAN)

-------------------------------------------------------------
```

Table 16-4: Parameters in the output of the **show bridge** command

Parameter	Meaning
Bridge Address	The MAC Address of the remote bridge, from which the Bridge Identifier used in the Spanning Tree Algorithm and Protocol is derived.
Bridge Name	The name of the bridge. This is the same as the value of the MIB object *sysDescr*.
Spanning Tree Protocol	Whether the Spanning Tree Protocol is enabled.
Filtering Learning	Whether filter learning is enabled.
Address Learning	Whether address learning is enabled.
Bridge Tagging (Local)	Whether the Local Bridge is enabled to receive IEEE 802 Tagged Frames (BCP Option 8 negotiation).
Bridge Tagging (Peer)	Whether the Peer Bridge is enabled to receive IEEE 802 Tagged Frames (BCP Option 8 negotiation).
Number of LAN Ports	The total number of LAN ports enabled for bridging.
VLAN Attached	The name of the VLAN attached to the Bridge or "-" if none.
Port Number	The number of the LAN port.
Port Address	The MAC address of the LAN port.
CAM	Whether CAM filtering is enabled for the interface.
Number Ports	The total number of WAN ports enabled for bridging.
Port Numbers	The WAN port ID numbers.
Number of Groups	The total number of remote bridge groups to which the bridge belongs.

Table 16-4: Parameters in the output of the **show bridge** command (cont.)

Parameter	Meaning
Ageingtime database	Seconds after which a dynamic entry is removed from the filtering database.
Uptime	Seconds since the remote bridge was last reset or initialized. This is the same as the value of the MIB object *sysUpTime*.

Examples To display the current configuration of the bridge module, use the command:

```
sh brid
```

Related Commands **disable bridge**
enable bridge

show bridge counter

Syntax `SHow BRIDge COUnter [POrt=1..32]`

Description This command displays information regarding the forwarding counters associated with LAN or virtual ports (Figure 16-12 on page 16-52, Table 16-5 on page 16-53). If a port is specified, details for the port are displayed.

Figure 16-12: Example output from the **show bridge counter** command

```
Port Counter
---------------------------------------------------------
                        ppp10      fr0 (22)   eth0


01:Fr. In (Data)        0000000000 0000000000 0000000000
02:Fr. In (STP)         0000000000 0000000000 0000000000
03:Fr. for relaying     0000000000 0000000000 0000000000
04:M-Cast Frames        0000000000 0000000000 0000000000
05:Dis: Inactive        0000000000 0000000000 0000000000
06:Dis: Inactive (STP)  0000000000 0000000000 0000000000
07:Dis: STP Ignored     0000000000 0000000000 0000000000
08:Dis: Framing Unknown 0000000000 0000000000 0000000000
09:Dis: MAC Equal       0000000000 0000000000 0000000000
10:Dis: Filter Match    0000000000 0000000000 0000000000
11:Dis: For bridge int. 0000000000 0000000000 0000000000
12:Dis: Same port       0000000000 0000000000 0000000000
13:Dis: No Ports        0000000000 0000000000 0000000000
14:Dis: Rcv Disab (STP) 0000000000 0000000000 0000000000
15:Dis: Fil Match (STP) 0000000000 0000000000 0000000000
16:Dis: Same Port (STP) 0000000000 0000000000 0000000000
17:Dis: No Ports (STP)  0000000000 0000000000 0000000000
18:Dis: Port Closed     0000000000 0000000000 0000000000
19:Dis: MTU Exceeded 1  0000000000 0000000000 0000000000
20:Dis: MTU Exceeded 2  0000000000 0000000000 0000000000
21:Dis: MTU Exceeded 3  0000000000 0000000000 0000000000
22:Relay (non-STP)      0000000000 0000000000 0000000000
23:Relay (STP)          0000000000 0000000000 0000000000
24:Relay Single         0000000000 0000000000 0000000000
25:Relay Mult.          0000000000 0000000000 0000000000
26:Bridge Gr Addr (STP) 0000000000 0000000000 0000000000
27:Bridge Ignored (STP) 0000000000 0000000000 0000000000
28:Source NonForw (STP) 0000000000 0000000000 0000000000
29:Relay Single (STP)   0000000000 0000000000 0000000000
30:Relay Mult. (STP)    0000000000 0000000000 0000000000
31:Port Open            0000000000 0000000000 0000000000
32:Port Closed          0000000000 0000000000 0000000000
33:Port Open (STP)      0000000000 0000000000 0000000000
34:Port Closed (STP)    0000000000 0000000000 0000000000
35:Down Ignore (Demand) 0000000000 0000000000 0000000000
36:Relay Out            0000000000 0000000000 0000000000
37:Send Out             0000000000 0000000000 0000000000
38:Sanity Check 1       0000000000 0000000000 0000000000
39:Sanity Check 2       0000000000 0000000000 0000000000
```

Table 16-5: Parameters in the output of the **show bridge counter** command

Parameter	Meaning
Interface Name	The name of the interface associated with the bridge port. For Frame Relay interfaces, the DLC number appears in parentheses after the interface name.
01: Fr. In (Data)	The number of data frames received.
02: Fr. In (STP)	The number of STP protocol frames received.
03: Fr for relaying	The number of frames passed to the relaying process.
04: M-Cast Frames	The number of multicast frames (including broadcast) frames received.
05: Dis: Inactive	The number of data frames discarded because the bridge was not active.
06: Dis: Inactive (STP)	The number of STP protocol frames dropped because the bridge was not active. Not supported on ethernet ports.
07: Dis: STP Ignored	The number of STP protocol frames ignored because STP was not active.
08: Dis: Framing Unknown	The number of frames discarded by the bridge module because their frame type could not be determined. The bridge supports 802.2, ETH-II and SNAP frames.
09: Dis: MAC Equal	The number of frames discarded because their source and destination MAC addresses were identical.
10: Dis: Filter Match	The number of frames discarded because they matched an entry in the filtering database (STP disabled).
11: Dis: For bridge int.	The number of frames discarded because they were destined for an interface on the bridging router.
12: Dis: Same port	The number of frames discarded because the destination station was known to be on the same port as the originating station, therefore the bridge need not forward those frames (STP disabled).
13: Dis: No Ports	The number of frames discarded because they could/should not be forwarded via any bridge port (STP disabled).
14: Dis: Rcv Disab (STP)	The number of frames discarded because the port they were received on was not enabled (STP enabled).
15: Dis: Fil Match (STP)	The number of frames discarded because they matched an entry in the filtering database (STP enabled).
16: Dis: Same Port (STP)	The number of frames discarded because the destination station was known to be on the same port as the originating station (STP enabled).
17: Dis: No Ports (STP)	The number of frames discarded because they could/should not be forwarded via any of the bridge’s ports (STP enabled).
18: Dis: Port Closed	The number of frames discarded because the port they were to be transmitted on is closed.
19: Dis: MTU Exceeded 1	The number of frames discarded because their size was larger than the MTU of the port/interface they were to be transmitted on (Case 1).
20: Dis: MTU Exceeded 2	The number of frames discarded because their size was larger than the MTU of the port/interface they were to be transmitted on (Case 2).

Table 16-5: Parameters in the output of the **show bridge counter** command (cont.)

Parameter	Meaning
21: Dis: MTU Exceeded 3	The number of frames discarded because their size was larger than the MTU of the port/interface they were to be transmitted on (Case 3).
22: Relay (non-STP)	The number of frames relayed while STP was disabled.
23: Relay (STP)	The number of frames relayed while STP was enabled.
24: Relay Single	The number of frames relayed via a single port (STP disabled).
25: Relay Mult.	The number of frames relayed via multiple ports (STP disabled).
26: Bridge Gr Addr (STP)	The number of frames received for the bridge group address (STP enabled).
27: Bridge Ignored (STP)	The number of frames received addressed to one of the bridge’s interfaces (STP enabled).
28: Source NonForw (STP)	The number of inbound frames discarded because the source port was not forwarding (STP enabled).
29: Relay Single (STP)	The number of frames relayed via a single port (STP enabled).
30: Relay Mult (STP)	The number of frames relayed via multiple ports (STP enabled).
31: Port Open	The number of times the lower-layer interface has indicated that this bridge port is open and able to transmit and receive bridge data.
32: Port Closed	The number of times the lower-layer interface has indicated that this bridge port is closed.
33: Port Open (STP)	The number of times the lower-layer interface has indicated that this bridge port is open and able to transmit and receive STP protocol traffic.
34: Port Closed (STP)	The number of times the lower-layer interface has indicated that this bridge port is closed for STP protocol traffic.
35: Down Ignore (Demand)	The number of times a “port closed” indication has been ignored because this port is a demand port.
36: Relay Out	The number of frames relayed out over the port.
37: Send Out	The number of frames sent via the port that were not relayed data frames.
38: Sanity Check 1	Internal debugging counter.
39: Sanity Check 2	Internal debugging counter.

Examples To display the counter for port 2, use the command:

```
sh brid po=2 cou
```

show bridge counter

Syntax `SHow BRIDge COUnter [POrt=1..32]`

Description This command displays information regarding the forwarding counters associated with virtual ports (Figure 16-13 on page 16-55, Figure 16-14 on page 16-56, and Table 16-6 on page 16-56). If a virtual port is specified, details are displayed for it. To display information from switch ports, use the **show switch port counter** command on page 8-49 of Chapter 8, Switching.

Figure 16-13: Example output from the **show bridge counter** command for a WAN-to-WAN bridge

```
Port Counter
------------------------------------------------------------
Interface                ppp10      fr0 (22)   fr0 (1)
Virtual Port Number      1          2          3
1:Fr. In (Data)          0000000000 0000000000 0000000000
02:Fr. for relaying      0000000000 0000000000 0000000000
03:M-Cast Frames         0000000000 0000000000 0000000000
04:Dis: Inactive         0000000000 0000000000 0000000000
05:Dis: STP Ignored      0000000000 0000000000 0000000000
06:Dis: Framing Unknown  0000000000 0000000000 0000000000
07:Dis: MAC Equal        0000000000 0000000000 0000000000
08:Dis: Filter Match     0000000000 0000000000 0000000000
09:Dis: For bridge int.  0000000000 0000000000 0000000000
10:Dis: Same port        0000000000 0000000000 0000000000
11:Dis: No Ports         0000000000 0000000000 0000000000
12:Dis: Port Closed      0000000000 0000000000 0000000000
13:Dis: MTU Exceeded 1   0000000000 0000000000 0000000000
14:Dis: MTU Exceeded 2   0000000000 0000000000 0000000000
15:Dis: MTU Exceeded 3   0000000000 0000000000 0000000000
16:Relay (non-STP)       0000000000 0000000000 0000000000
17:Relay Single          0000000000 0000000000 0000000000
18:Relay Mult.           0000000000 0000000000 0000000000
19:Port Open             0000000000 0000000000 0000000000
20:Port Closed           0000000000 0000000000 0000000000
21:Down Ignore (Demand)  0000000000 0000000000 0000000000
22:Relay Out             0000000000 0000000000 0000000000
23:Send Out              0000000000 0000000000 0000000000
24:Sanity Check 1        0000000000 0000000000 0000000000
25:Sanity Check 2        0000000000 0000000000 0000000000
------------------------------------------------------------
```

Figure 16-14: Example output from the **show bridge counter** command for a VLAN-to-WAN bridge.

```
Port Counter
------------------------------------------------------------
Interface                vlan      fr0 (22)
Virtual Port Number      -          1
1:Fr. In (Data)          0000000000 0000000000
02:Fr. for relaying      0000000000 0000000000
03:M-Cast Frames         0000000000 0000000000
04:Dis: Inactive         0000000000 0000000000
05:Dis: STP Ignored      0000000000 0000000000
06:Dis: Framing Unknown  0000000000 0000000000
07:Dis: MAC Equal        0000000000 0000000000
08:Dis: Filter Match     0000000000 0000000000
09:Dis: For bridge int.  0000000000 0000000000
10:Dis: Same port        0000000000 0000000000
11:Dis: No Ports         0000000000 0000000000
12:Dis: Port Closed      0000000000 0000000000
13:Dis: MTU Exceeded 1   0000000000 0000000000
14:Dis: MTU Exceeded 2   0000000000 0000000000
15:Dis: MTU Exceeded 3   0000000000 0000000000
16:Relay (non-STP)       0000000000 0000000000
17:Relay Single          0000000000 0000000000
18:Relay Mult.           0000000000 0000000000
19:Port Open             0000000000 0000000000
20:Port Closed           0000000000 0000000000
21:Down Ignore (Demand)  0000000000 0000000000
22:Relay Out             0000000000 0000000000
23:Send Out              0000000000 0000000000
24:Sanity Check 1        0000000000 0000000000
25:Sanity Check 2        0000000000 0000000000
------------------------------------------------------------
```

Table 16-6: Parameters in the output of the **show bridge counter** command

Parameter	Meaning
Interface Name	The name of the interface associated with the bridge port.
Port Number	The virtual port number for the interface.
01: Fr. In (Data)	The number of data frames received.
02: Fr for relaying	The number of frames passed to the relaying process.
03: M-Cast Frames	The number of multicast frames (including broadcast) frames received.
04: Dis: Inactive	The number of data frames discarded because the bridge was not active.
05: Dis: STP Ignored	The number of STP protocol frames ignored because STP was not active.
06: Dis: Framing Unknown	The number of frames discarded by the bridge module because their frame type could not be determined. The bridge supports 802.2, ETH-II and SNAP frames.
07: Dis: MAC Equal	The number of frames discarded because their source and destination MAC addresses were identical.
08: Dis: Filter Match	The number of frames discarded because they matched an entry in the filtering database (STP disabled).
09: Dis: For bridge int.	The number of frames discarded because they were destined for an interface on the bridge.

Table 16-6: Parameters in the output of the **show bridge counter** command (cont.)

Parameter	Meaning
10: Dis: Same port	The number of frames discarded because the destination station was known to be on the same port as the originating station, therefore the bridge need not forward those frames.
11: Dis: No Ports	The number of frames discarded because they could/should not be forwarded via any bridge port.
12: Dis: Port Closed	The number of frames discarded because the port they were to be transmitted on is closed.
13: Dis: MTU Exceeded 1	The number of frames discarded because their size was larger than the MTU of the port/interface they were to be transmitted on (Case 1).
14: Dis: MTU Exceeded 2	The number of frames discarded because their size was larger than the MTU of the port/interface they were to be transmitted on (Case 2).
15: Dis: MTU Exceeded 3	The number of frames discarded because their size was larger than the MTU of the port/interface they were to be transmitted on (Case 3).
16: Relay	The number of frames relayed.
17: Relay Single	The number of frames relayed via a single port.
18: Relay Mult.	The number of frames relayed via multiple ports.
19: Port Open	The number of times the lower-layer interface has indicated that this bridge port is open and able to transmit and receive bridge data.
20: Port Closed	The number of times the lower-layer interface has indicated that this bridge port is closed.
21: Down Ignore (Demand)	The number of times a “Port Closed” indication has been ignored because this port is a demand port.
22: Relay Out	The number of frames relayed out over the port.
23: Send Out	The number of frames sent via the port that were not relayed data frames.
24: Sanity Check 1	Internal debugging counter.
25: Sanity Check 2	Internal debugging counter.

Examples To display the counters for virtual port 2, use the command:

```
sh brid po=2 cou
```

Related Commands

`show bridge port`
`show bridge`
`show bridge counter`

show bridge filter

Syntax

```
SHow BRIDge FILter[=1..99] [ENTry=entry]
```

where *entry* is a filter number from 1 to *n*, and *n* is the number of filter entries currently defined in the filter

Description

This command displays information about one or all bridge filters, or one of the entries in a bridge filter (Figure 16-15 on page 16-59, Table 16-7 on page 16-59).

The **filter** parameter specifies the bridge filter to be displayed. The specified filter must exist. If no filter is specified, all filters are displayed.

The **entry** parameter specifies a particular entry in the filter. The specified filter entry must exist in the filter. If the **entry** parameter is specified, the **filter** parameter must specify a valid filter number.

The counters given in the output are related in the following ways:

- "Frames seen" = "Frames passed" + "Frames dropped".
- "Frames seen" = "Frames unmatched" + sum of "Matches".
- "Frames passed" = sum of "Matches" for entries for which "Output ports" is not "None".
- "Frames dropped" = "Frames unmatched" + sum of "Matches" for entries for which "Output ports" is "None".

Whenever an entry is added to a bridge filter, counters are not cleared. However, when an entry is modified or deleted, the "Matches" count for the entry no longer reflects the frames matched by that filter, so the "Matches" count is cleared to 0. The "Frames seen" counter and one of the "Frames dropped" or "Frames passed" counters (based on the previous value of "Output ports" for the entry) are decremented by the value of "Matches" so that the relationships are maintained.

Figure 16-15: Example output from the **show bridge filter** command

```
Bridge filters
------------------------------------------------------------
Filter .............. 1
Used by ports........ None
Frames seen ......... 37465
Frames passed ....... 4938
Frames unmatched .... 2652
Frames dropped ...... 32527

Entry ............... 1
Source address ........ = 00-00-cd-00-00-00/ff-ff-ff-00-00-00
Dest address .......... Match any
Protocol ......... .... = ETHII, = 0800
Size .................. Match any
Multicast types ....... Match any
Output ports .......... 1,2
Matches ............... 4938
Entry ............... 2
Source address ........ Match any
Dest address .......... Match any
Protocol .............. = ETHII
Size .................. Match any
Multicast types ....... Match any
Data Offset ........... 27
Data Pattern .......... = 345678
Output ports ........ . None
Matches .......... .... 29875
------------------------------------------------------------
```

Table 16-7: Parameters in the output of the **show bridge filter** command

Parameter	Meaning
Filter	The filter number for this filter.
Used by ports	The list of ports that are currently using this filter.
Frames seen	The number of frames to which this filter has been applied.
Frames passed	The number of frames passed by this filter.
Frames unmatched	The number of frames for which a filter entry match was not made. These frames are dropped and included in the Frames dropped count.
Frames dropped	The number of frames dropped by this filter.
Entry	The filter number for the filter entry.
Source address	The condition, address and mask for matching source addresses for the filter entry.
Dest address	The condition, address and mask for matching destination addresses for the filter entry.
Protocol	The condition, Ethernet encapsulation and discriminator for the filter entry.
Size	The condition and size of frame for the filter entry.
Multicast types	The condition and multicast frame types for the filter entry.
Data Offset	The offset in the data field of the DATA condition specified in the Data Pattern field.

Table 16-7: Parameters in the output of the **show bridge filter** command (cont.)

Parameter	Meaning
Data Pattern	The condition for the data in the data field starting at the position specified in the Data Offset field.
Output ports	The list of output ports for the filter entry.
Matches	The number of times this filter entry has matched a bridged frame. If the output ports field is “None”, matches are included in the Frames dropped count. Otherwise, matches are included in the Frames passed count.

Examples

To display information about all bridge filters, use the command:

```
sh brid fil
```

Related Commands

add bridge filter
delete bridge filter
set bridge filter

show bridge group

Syntax `SHow BRIDge GROup[={groupname|1..32}]`

where *groupname* is a unique name for the group up to 32 characters long

Description This command displays information regarding the bridge's attachment to specific groups (Figure 16-16 on page 16-61, Table 16-8 on page 16-61). The group may be specified by either the group number or name. If a group is specified, information is displayed for it, otherwise information is displayed for all groups.

Figure 16-16: Example output from the **show bridge group** command

```
------------------------------------------------------------
Group Number                 : 1
Group Name                   : Branch1 - Head Office
Virtual Port(s)              : 3, 4
Reclustering                 : FALSE
Current Cluster Identifier : A21B : 00-00-0C-00-00-1A
Old Cluster Identifier       : A21B : 00-00-0C-00-00-1C
Reclustering Delay           : 4
Primary Reclustering Delay : 4

Group Number                 : 2
Group Name                   : Branch2 - Accounts
Virtual Port                 : 2
Reclustering                 : FALSE
Current Cluster Identifier : A21B : 00-00-0C-00-00-2d
Old Cluster Identifier       : A21B : 00-00-0C-00-00-79
Reclustering Delay           : 4
Primary Reclustering Delay : 4
------------------------------------------------------------
```

Table 16-8: Parameters in the output of the **show bridge group** command

Parameter	Meaning
Group Number	The assigned number for the group.
Group Name	The name of the group.
Virtual Port(s)	The virtual port numbers of the ports belonging to the group.
Reclustering	A boolean parameter, set to TRUE if and only if a Reclustering Delay period applies to the remote bridge's virtual ports attaching to this group. Always set to FALSE for a Virtual LAN.
Current Clustering Identifier	The cluster identifier of the Cluster, if any, to which the remote bridge considers itself to belong within the Group. If the bridge does not consider it belongs to any cluster this is set NULL.
Old Cluster Identifier	When the Reclustering parameter is FALSE, this parameter takes the same value as the Current Cluster Identifier parameter. When the Reclustering parameter is TRUE, this parameter takes the value to which the Current Cluster Identifier parameter was set at the time when the Reclustering parameter was last set to FALSE.

Table 16-8: Parameters in the output of the **show bridge group** command (cont.)

Parameter	Meaning
Reclustering Delay	The time period for which the bridge waits, following detection of a possible reclustering, before selecting a new cluster membership.
Primary Reclustering Delay	The value of the Reclustering Delay parameter to be used when the bridge is the primary bridge for a cluster in the group.

Examples To display all bridge groups, use the command:

```
sh brid gro
```

Related Commands

add bridge group
delete bridge group
set bridge group

show bridge port

Syntax `SHow BRIDge POrt[=1..32]`

Description This command displays general information about LAN Ports or virtual ports. Spanning tree information is displayed if the Spanning Tree Protocol is enabled on the port (Figure 16-17 on page 16-63 and Figure 16-18 on page 16-64, Table 16-9 on page 16-64). If a port is specified, information is displayed for the specified port; otherwise, information is displayed for all ports.

Figure 16-17: Example output from the **show bridge port** command for a port with the Spanning Tree Protocol disabled

```
Port Information
------------------------------------------------------------
Port Number                     : 1
Port Interface                  : eth0
Port Media Type                 : ISO8802-3 CSMACD PPP
Port filter                     : 2
Outbound User Priority          : 0
Outbound Access Priority        : 0
UpTime                          : 0

Port Number                     : 2
Port Name                       : wan
Port Interface                  : PPP0
Port Media Type                 : PPP
Port filter                     : -
Outbound User Priority          : 0
Outbound Access Priority        : 0
UpTime                          : 0
------------------------------------------------------------
Spanning Tree Protocol disabled, STP information not shown.
```

Figure 16-18: Example output from the **show bridge port** command for a port with the Spanning Tree Protocol enabled

```
Port Information
------------------------------------------------------------
Port Number                   : 1
Port Interface                : eth0
Port Media Type               : ISO8802-3 CSMACD
Port filter                   : 2
Outbound User Priority        : 0
Outbound Access Priority      : 0
UpTime                        : 0
State                         : forwarding
Port Identifier               : 8001
Path Cost                     : 100
Designated Root               : 8000 : 00-00-cd-00-0f-29
Designated Cost               : 0
Designated Bridge             : 8000 : 00-00-cd-00-0f-29
Designated Port               : 8001
Topology Change Acknowledge   : 0

Port Number                   : 2
Port Interface                : fr0
Port Media Type               : FRAMERELAY
Port filter                   : -
Outbound User Priority        : 0
Outbound Access Priority      : 0
UpTime                        : 0
State                         : forwarding
Port Identifier               : 8002
Path Cost                     : 100
Designated Root               : 8000 : 00-00-cd-00-0f-29
Designated Cost               : 0
Designated Bridge             : 8000 : 00-00-cd-00-0f-29
Designated Port               : 8002
Topology Change Acknowledge   : 0
------------------------------------------------------------
```

Table 16-9: Parameters in the output of the **show bridge port** command

Parameter	**Meaning**
Port Number	The number of the port.
Port Interface	The interface name for the port. This is the same as the value of the MIB object ifDescr.
Port Media Type	The MAC entity type as defined in the MIB object ifType.
Port filter	The bridge filter, if any, defined for this port.
Outbound User Priority	The outbound user priority.
Outbound Access Priority	The outbound access priority.
Uptime	The count in seconds of the elapsed time since the port was last reset or initialized.
State	The current state of the bridge port; either disabled, listening, learning, forwarding, or blocking.

Table 16-9: Parameters in the output of the **show bridge port** command (cont.)

Parameter	Meaning
Port Identifier	This parameter comprises two parts. One part being the actual port number, the other more significant number is the priority of the port.
Path Cost	The contribution of the path through the port, when the Port is the Root Port, to the total cost of the path to the Root for this bridge.
Designated Root	The unique Bridge Identifier of the bridge recorded as the Root in the Root Identifier parameter of Configuration Messages transmitted by the Designated Bridge on the LAN or Subgroup to which the Port is attached.
Designated Cost	For a Designated Port, the path cost offered to the LAN or Subgroup to which the Port is attached; otherwise, the cost of the path to the Root offered by the Designated Port on the LAN or Subgroup to which this port is attached.
Designated Bridge	The unique Bridge Identifier of for a Designated Port, the bridge to which the Port belongs, or otherwise, the bridge believed to be the Designated Bridge for the LAN or Subgroup to which this port is attached.
Designated Port	The Port Identifier of the bridge port, on the Designated Bridge, through which the Designated Bridge transmits the configuration message information stored by this port.
Topology Change Acknowledge	The value of the Topology Change Acknowledgment flag in the next configuration message to be transmitted via the port.

Examples To display the configuration for bridge port 1, use the command:

```
sh brid po=1
```

Related Commands

add bridge port
delete bridge port
set bridge port

show bridge protocol

Syntax `SHow BRIDge PROTocol`

Description This command displays information about the protocols that are currently enabled for bridging (Figure 16-19, Table 16-10).

Figure 16-19: Example output from the **show bridge protocol** command

```
Index   Encapsulation   Protocol     Name     Priority
-------------------------------------------------------
1       ETHII           6004         LAT         1
2       ETHII           6003         DECnet      2
200     SNAP            080007809b               4
-------------------------------------------------------
```

Table 16-10: Parameters in the output of the **show bridge protocol** command

Parameter	Meaning
Index	A manager-defined index. If no index is given, one is assigned.
Encapsulation	The encapsulation of the frame; either "EthII" (IEEE 802.3), "SAP" (IEEE 802.2) standard with SAPs, "SNAP" (IEEE 802.2 using the SNAP mechanism), or "Novell" (original Novell).
Protocol	The actual protocol field.
Name	The descriptive name, if any, assigned when the protocol was added.
Priority	The forwarding priority assigned to the protocol; either "0" (lowest), "1" (default), "2", "3" or "4" (highest).

Examples To display the list of protocols being bridged, use the command:

```
sh brid prot
```

Related Commands **add bridge protocol**
delete bridge protocol

show bridge spanning

Syntax `SHow BRIDge SPanning`

Description This command displays information regarding the Spanning Tree Protocol operating status and settings (Figure 16-20 on page 16-67, Table 16-11 on page 16-67).

Figure 16-20: Example output from the **show bridge spanning** command

```
Spanning Tree Protocol
------------------------------------------------------------
Bridge Identifier            : 8000 : 00-00-cd-00-0d-4d
Time Since Topology Change   : 0
Topology Change Count        : 0
Topology Change              : 0
Designated Root              : 8000 : 00-00-cd-00-0d-4d
Root Port                    : 0
Root Path Cost               : 0
Max Age                      : 40
Hello Time                   : 2
Forward Delay                : 15
Bridge Max Age               : 40
Bridge Hello Time            : 2
Bridge Forward Delay         : 15
Hold Time                    : 1
------------------------------------------------------------
```

Table 16-11: Parameters in the output of the **show bridge spanning** command

Parameter	Meaning
Bridge Identifier	The unique Bridge Identifier of the bridge. This parameter consists of two parts, one of which is derived from the unique Bridge Address, and the other of which is the priority of the bridge.
Time Since Topology Change	Seconds elapsed since the Topology Change parameter last happened.
Topology Change Count	The number of times the Topology Change parameter has been set since the remote bridge was initialized.
Topology Change	This parameter is used to propagate the indication of topology change in transmitted Configuration Messages, and to determine whether short or long timeout values are to be used for dynamic entries in the Filtering Database.
Designated Root	The unique Bridge identifier of the bridge assumed to be the Root.
Root Port	The port number of the root port for the bridge, or 0 if this is the Root.
Root Path Cost	The cost of the path to the Root from this bridge. When the bridge is the Root this parameter has the value zero.
Max Age	The maximum age of received Configuration Message information before it is discarded.

Table 16-11: Parameters in the output of the **show bridge spanning** command (cont.)

Parameter	**Meaning**
Hello Time	The time interval between successive transmissions of the Configuration Message information by a bridge that is attempting to become the Root or which is the Root.
Forward Delay	The time spent by a Port in the Listening state and Learning state before moving to the Learning or Forwarding state respectively. Also the value used for the ageing time of the dynamic entries in the Filtering Database while received Configuration Messages indicate a topology change.
Bridge Max Age	The value of the Max Age parameter when the bridge is the Root or is attempting to become the Root. This parameter can be set by Management.
Bridge Hello Time	The value of the Hello Time parameter when the bridge is the Root or is attempting to become the Root. This parameter can be set by Management.
Bridge Forward Delay	The value of the Forward Delay parameter when the bridge is the Root or is attempting to become the Root. This parameter can be set by Management.
Hold Time	The minimum time period to elapse between the transmission of configuration BPDUs through a given LAN Port. This parameter is a fixed parameter, with value as specified in IEEE Std 802.1D [2].

Examples To display the current configuration of the Spanning Tree Protocol, use the command:

```
sh brid sp
```

Related Commands

disable bridge spanning
enable bridge spanning
set bridge spanning

show bridge station

Syntax `SHow BRIDge STation [{Address=`*`macadd`*` [MASK=`*`macadd`*`]|`
`POrt=1..32}]`

where *macadd* is an Ethernet six-octet MAC address, expressed as six pairs of hexadecimal digits delimited by hyphens

Description This command displays the bridge module bridge station map (Figure 16-21 on page 16-69, Table 16-12 on page 16-70). The bridge station map records that port should be used to transmit frames to all Ethernet MAC addresses the bridge module knows about.

The bridge station map and associated learning is inactive in the VLAN-to-WAN mode. Running this command while configured for Remote VLAN operation, is likely to display inaccurate configuration information.

The **address** parameter specifies a particular MAC address to display, and limits the display to entries with this address after the address has been ANDed with the optional **mask**. If **address** is not specified, all bridge station map entries are displayed.

The **mask** parameter specifies a MAC address mask to widen the range of entries to display. The address of an entry in the dynamic bridge station map is ANDed with the mask and compared to the address given with the **address** parameter. If there is a match, the entry is displayed. The default is **ff-ff-ff-ff-ff-ff**.

The **port** parameter specifies a bridge port name or number for which bridge station map entries are to be displayed.

Figure 16-21: Example output from the **show bridge station** command

```
MAC address          Type          Port
---------------------------------------
00-00-c0-0e-26-f8    Learned        1
00-00-c0-c9-c6-7b    Learned        1
01-80-c2-00-00-10    self           0
01-80-c2-00-00-0f    self           0
01-80-c2-00-00-0e    self           0
01-80-c2-00-00-0d    self           0
01-80-c2-00-00-0c    self           0
01-80-c2-00-00-0b    self           0
01-80-c2-00-00-0a    self           0
01-80-c2-00-00-09    self           0
01-80-c2-00-00-08    self           0
01-80-c2-00-00-07    self           0
01-80-c2-00-00-06    self           0
01-80-c2-00-00-05    self           0
01-80-c2-00-00-04    self           0
01-80-c2-00-00-03    self           0
01-80-c2-00-00-02    self           0
01-80-c2-00-00-01    self           0
01-80-c2-00-00-00    self           0
00-00-cd-00-2c-a0    self           0
---------------------------------------
```

Table 16-12: Parameters in the output of the **show bridge station** command

Parameter	Meaning	
MAC address	The MAC address for this entry in the bridge station map.	
Type	The type of bridge station map entry:	
	Self	addresses that the bridge itself receives frames on
	Management	entries added with the **add bridge station** command on page 16-33 or by SNMP
	Learned	addresses learned as part of the filtering and forwarding process
Port		The bridge port index.

Related Commands **add bridge station**
delete bridge station