

Chapter 17

Bridging

Introduction	17-2
Bridging on the Switch	17-2
Remote Bridging	17-3
Virtual Ports and Switch Ports	17-3
VLAN-to-WAN Bridging	17-5
Internal Representation of the VLAN-to-WAN Bridge	17-5
The VLAN-to-WAN Bridging Process	17-5
WAN-to-WAN Bridging	17-6
Internal Representation of the WAN-to-WAN Bridge	17-7
Bridge Learning and Forwarding	17-7
Configuration Examples	17-9
VLAN-to-WAN Bridge Configuration	17-9
Command Reference	17-11
add bridge filter	17-11
add bridge port	17-14
add bridge station	17-15
add vlan bridge	17-16
delete bridge filter	17-17
delete bridge port	17-17
delete bridge station	17-18
delete vlan bridge	17-18
disable bridge	17-19
disable bridge learning	17-19
disable bridge tagged	17-19
enable bridge	17-20
enable bridge learning	17-20
enable bridge tagged	17-20
purge bridge	17-21
reset bridge	17-21
set bridge ageingtimer	17-21
set bridge filter	17-22
set bridge port	17-24
show bridge	17-25
show bridge counter	17-27
show bridge filter	17-30
show bridge port	17-33
show bridge protocol	17-34
show bridge station	17-35

Introduction

This chapter describes:

- the bridging function on the switch
- how bridging is implemented on the switch
- how to configure and operate the switch in one of the following modes:
 - VLAN-to-WAN bridging
This configuration enables traffic from a single VLAN to be shared across a Frame Relay wide area network or 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 17-5](#).
 - WAN-to-WAN bridging
This configuration bridges traffic between two or more virtual ports to link remote WAN connected LANs so as to forward layer two traffic between them. See [“WAN-to-WAN Bridging” on page 17-6](#).

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 switch to be reset.
- Management of the bridge station map.
- Learning MAC addresses.
- Filtering and forwarding packets. It accepts all packets on its interfaces and forwards them with no distinction between protocols.
- Support for virtual ports over point-to-point (PPP) and Frame Relay interfaces.
- Support for on-demand ports over PPP interfaces.
- Operation of configurable bridge filters that further modify the filtering and forwarding processes.

Attachment to a VLAN to provide Remote Bridging.

Bridging on the Switch

The implementation of the bridge module in the switch 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 switch 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 7, and [“WAN-to-WAN Bridging” on page 17-6](#).

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 17-6](#).

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 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, or a Frame Relay virtual circuit.

Note that spanning tree protocol (STP) is supported over switch ports (see [Chapter 8, Switching](#)), but is not supported over the bridge's virtual ports.

Table 17-1: 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.

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 switch may need careful configuration (the addition of filters, for example) to avoid unnecessary calls.

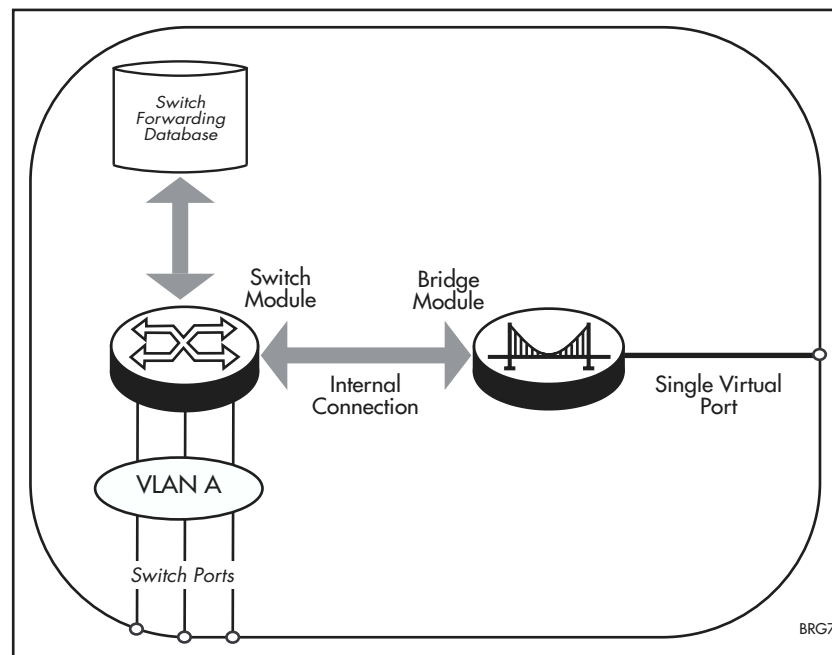
VLAN-to-WAN Bridging

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.

Internal Representation of the VLAN-to-WAN Bridge

Figure 17-1 on page 17-5 shows an internal representation of the VLAN-to-WAN bridge that exists on the Rapier Series switches. Note that the bridge and switch symbols are internal functional representations and are not standalone devices.

Figure 17-1: Internal representation of the switch's VLAN-to-WAN bridge



Switch ports within each VLAN connect to the switch module, to obtain layer two connectivity (local or remote) for their attached devices. An internal data path, shown by the horizontal grey arrow, provides connectivity between the two modules.

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 17-1 on page 17-5](#), provides connectivity between the two modules.

Because the VLAN-to-WAN bridge has one virtual port and all traffic passes between it and the switch module, there is no need for the bridge station map to learn addresses. However, although there is a station learning process within this configuration that is handled by the switch forwarding database. For more information, see [Chapter 8, Switching](#).

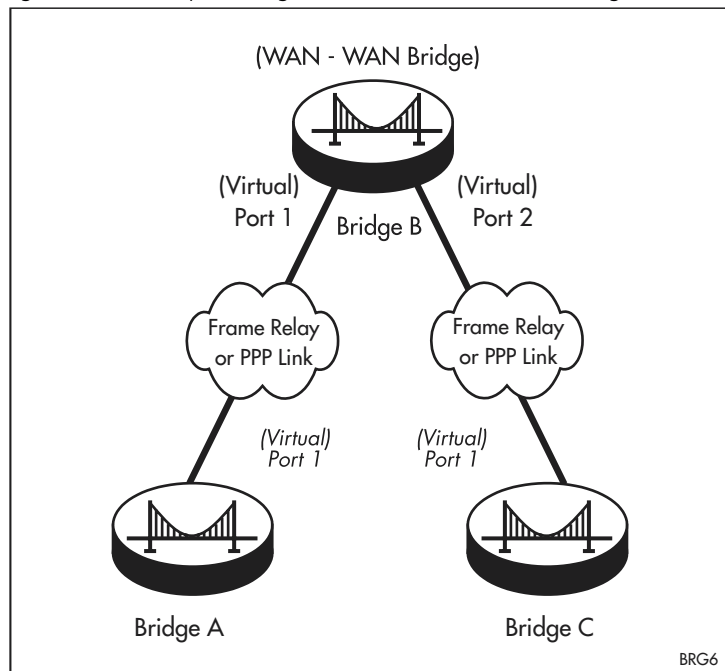
Data frames cross the wide area link untagged and are retagged by the VLAN-to-WAN bridge. Frames entering the bridge from the wide area network are assigned the VLAN tag that the bridge associates with the link. However, the forwarding process itself is handled by the switch forwarding database. For more information, see [Chapter 8, Switching](#).

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 17-2 on page 17-6](#) below shows a possible WAN-to-WAN bridge configuration.

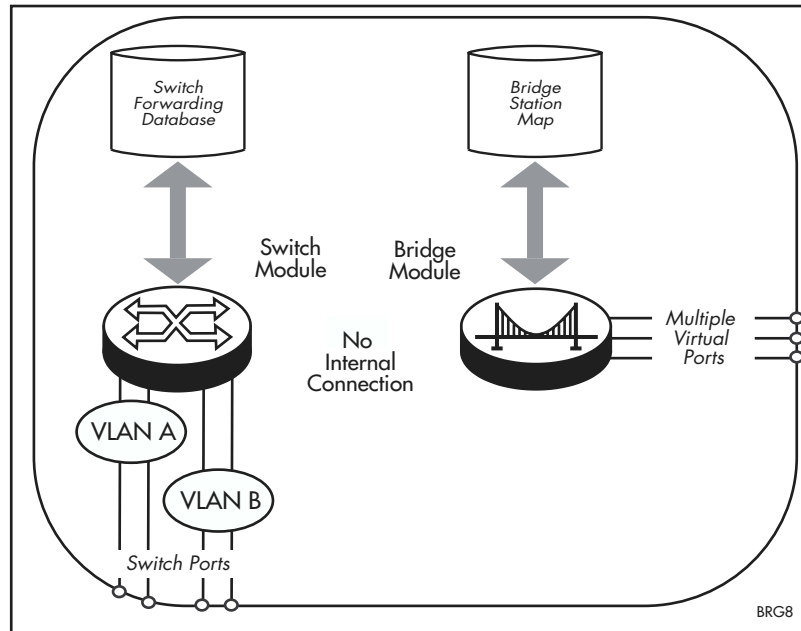
Figure 17-2: Example configuration of a WAN-to-WAN bridge



Internal Representation of the WAN-to-WAN Bridge

Figure 17-3 on page 17-7 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 17-3: 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](#).

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.

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 frame's 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.

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 following example shows how to configure an extended LAN.

VLAN-to-WAN Bridge Configuration

Figure 17-4 on page 17-9 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 17-4: Example configuration for a remotely bridged VLAN

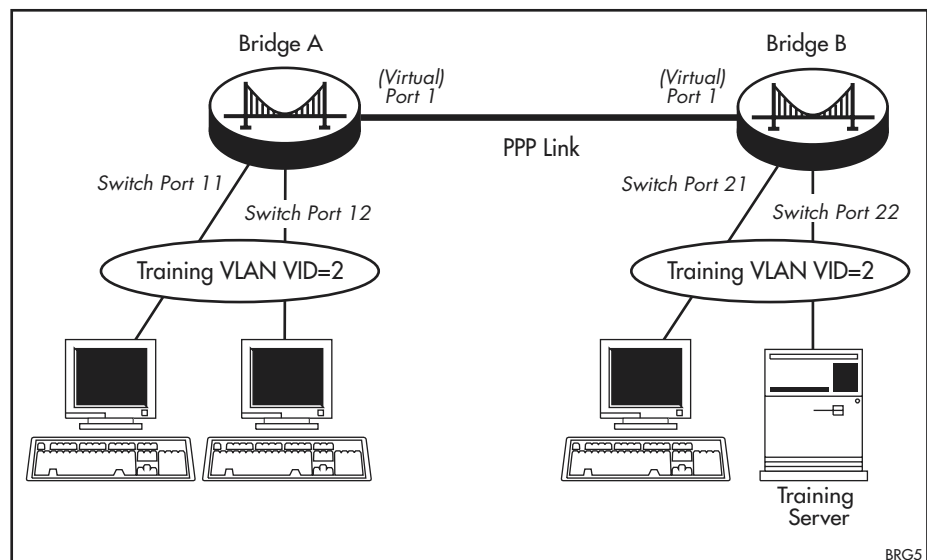


Table 17-2: VLAN membership of example of a network using tagged ports

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

To configure VLAN-to-WAN bridge A

1. 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
```

2. Add switch ports to the VLAN.

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

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

3. Add the VLAN to the bridge.

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

```
add vlan=2 bridge
```

4. Create a WAN interface.

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

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

5. 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. 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
```

2. Add switch ports to the VLAN.

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

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

3. Add the VLAN to the bridge.

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

```
add vlan=2 bridge
```

4. Create a WAN interface.

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

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

5. 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 and PPP, see [Chapter 15, Frame Relay](#), and [Chapter 16, Point-to-Point Protocol \(PPP\)](#).

Command Reference

This section describes the commands available on the switch 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 lxvi 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.

Note that these filters operate on the virtual ports on ingress, and determine the ports where each frame is forwarded. Filtering on switch ports is an entirely separate and simultaneously process. See [Chapter 8, Switching](#) for information about filtering on switch ports.

The **port** parameter specifies the virtual 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 virtual ports. Only a single virtual port may be configured to a switch operating in the remote VLAN mode. If **none** is specified, the frame may not be forwarded, and should be discarded. If a comma-separated list of virtual 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. Switch ports are unaffected by this command.

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. 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 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 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 interface to be added to a virtual port. Valid interfaces are:

- 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 used when connecting to a Frame Relay interface. It specifies the Frame Relay DLCI that is assigned to the port.

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

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

To add DLC23 on Frame Relay interface 1 to the bridge as virtual port 2, use the command:

```
add brid po=2 int=fr1 circ=23
```

Related Commands

- [delete bridge port](#)
- [set bridge port](#)
- [show bridge port](#)

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 virtual 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 virtual port out which the station is found. The specified virtual port must exist.

Examples To add a bridge station map entry for MAC address 00-00-cd-12-34-56, which is reached via virtual 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

- 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 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. Note that only a single VLAN can be attached to the bridge.

For Frame Relay operation, only a single VLAN can be attached to the bridge.

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

```
add vlan=training bridg
```

Related Commands

- [add bridge port](#)
- [delete vlan bridge](#)
- [enable bridge](#)
- [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 port

Syntax `DELeTe BRIDGe PORt=1..32`

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

Examples To delete bridge virtual port 2, use the command:

```
del brid po=2
```

Related Commands [add bridge port](#)
[set bridge port](#)
[show bridge port](#)

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 virtual 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 virtual 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 virtual 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](#)
[show vlan](#)

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 bridge learning is disabled and the ageing timer has aged out all dynamically learned entries, statically entered MAC source addresses are used to decide the packets to forward or discard. If no matching entries are found in the bridge station map during the forwarding process, then all virtual ports on the bridge are flooded with the frame, except the port where the frame was received.

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

```
dis brid le
```

Related Commands [enable bridge learning](#)
[show bridge](#)

disable bridge tagged

Syntax DISable BRIDge TAGged

Description This command disables BCP Option 8 negotiation (IEEE 802 Tagged Frames) with the Peer at the other end of the bridge.

Examples To disable BCP Option 8 negotiation use the command:

```
dis brid tag
```

Related Commands [enable bridge tagged](#)
[show bridge](#)

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 tagged

Syntax ENAbLe BRIDgE TAGged

Description This command enables BCP Option 8 negotiation (IEEE 802 Tagged Frames) with the Peer at the other end of the bridge.

Examples To enable BCP Option 8 negotiation use the command:

```
ena brid tag
```

Related Command [disable bridge tagged](#)
[show bridge](#)

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. 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 virtual 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 virtual ports. If **none** is specified, the frame may not be forwarded, and should be discarded. If a comma-separated list of virtual ports is specified, the frame forwarding procedure decides which of the specified virtual ports should receive the frame based upon its analysis of the bridged traffic. The command has no effect on switch ports.

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 port

Syntax SET BRIDGE PORT=1..32 [FILTER={NONE|1..99}]

Description This command sets or changes the filters associated with a specified port.

The **filter** parameter specifies the bridge filter to be associated with this port. The value **none** is used to turn off bridge filtering for frames received on this port. The values 1 to 99 specify a bridge filter that filters all frames received on this port. By default, no bridge filter is associated with a port.

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

```
set bridge po=2 fil=2
```

Related Commands [add bridge port](#)
[delete bridge port](#)
[show bridge port](#)

show bridge

Syntax SHow BRIDge

Description This command displays configuration information for the bridge ([Figure 17-5](#), [Table 17-3 on page 17-25](#)).

Figure 17-5: 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          : Switch 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)
-----

```

Table 17-3: Parameters in the output of the **show bridge** command

Parameter	Meaning
Bridge Address	The MAC Address of the remote bridge.
Bridge Name	The name of the bridge. This is the same as the value of the MIB object <i>sysDescr</i> .
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).
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.
Number of virtual Ports	The total number of virtual or WAN ports enabled for bridging.
Virtual Port Numbers	The virtual or WAN port ID numbers.
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 <i>sysUpTime</i> .

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

```
sh brid
```

Related Commands

- [disable bridge](#)
- [disable bridge tagged](#)
- [enable bridge](#)
- [enable bridge tagged](#)

show bridge counter

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

Description This command displays information regarding the forwarding counters associated with virtual ports ([Figure 17-6 on page 17-27](#), [Figure 17-7 on page 17-28](#), and [Table 17-4 on page 17-28](#)). 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 8-139 of Chapter 8, Switching](#).

Figure 17-6: 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 17-7: 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 17-4: 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.
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).

Table 17-4: Parameters in the output of the **show bridge counter** command (cont.)

Parameter	Meaning
09: Dis: For bridge int.	The number of frames discarded because they were destined for an interface on the bridge.
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 17-8 on page 17-31](#), [Table 17-5 on page 17-31](#)).

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 17-8: Example output from the **show bridge filter** command

```

Bridge filters
-----
Filter ..... 1
Used by virtual 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 17-5: Parameters in the output of the **show bridge filter** command

Parameter	Meaning
Filter	The filter number for this filter.
Used by virtual 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 17-5: 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 virtual 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 port

Syntax `SHoW BRIDgE POrt [=1..32]`

Description This command displays general information about the virtual ports (see [Table 17-9 on page 17-33](#)). If a port is specified, information is displayed for the specified virtual port; otherwise, information is displayed for all virtual ports.

Figure 17-9: Example output from the **show bridge port** command

Port Information	

Port Number	: 1
Port Interface	: PPP1
Port Media Type	: PPP
Port filter	: 2
UpTime	: 0
Port Number	: 2
Port Interface	: PPP0
Port Media Type	: PPP
UpTime	: 0

Table 17-6: 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.
Uptime	The count in seconds of the elapsed time since the port was last reset or initialized.

Examples To display the configuration for bridge virtual 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 17-10, Table 17-7).

Figure 17-10: 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 17-7: 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
```

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 17-11 on page 17-35, Table 17-8 on page 17-36). The bridge station map records that virtual 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 virtual port name or number for which bridge station map entries are to be displayed.

Figure 17-11: 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 17-8: 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 17-15 or by SNMP Learned addresses learned as part of the filtering and forwarding process
Virtual Port	The virtual port ID number.

Related Commands [add bridge station](#)
[delete bridge station](#)