Chapter 42

Xbar Switches - Filtering and Forwarding

TEC/NOT/069

Switches are a critical component in any networked FTI data acquisition system in order to allow the forwarding of data from the Data Acquisition Unit (DAU) to the target destination devices such as the network recorder, Ethernet-to-PCM gateway or ground station.

This paper describes the fully interconnected Xbar switch fabric technology implemented in the Curtiss-Wright Xbar switch product family. The Xbar technology is designed to meet the reliability and performance requirements of FTI equipment whilst providing full flexibility and configurability to meet any topological requirements.

This paper discusses the following topics:

NOTE: It is assumed that the reader is familiar with network switching and the Simple Network Management Protocol (SNMP). To better understand this paper, see "42.6 Recommended reading" on page 23.

42.1 Overview of network switches

In an FTI network of distributed peer DAUs, the switch is a key component that allows data to be transmitted to and from different nodes in the network. Switches are comprised of a number of ports to which DAUs may be connected or which can be used to interconnect other switches. In general, connections in the switch utilize point-to-point full-duplex Ethernet links. The most important task of the switch is to reliably and quickly forward and route packets to their destination. Quite often the terms routing and forwarding are used interchangeably. There is in fact a subtle relationship between the two, see "42.1.0.1 Routing" on page 1 and "42.1.0.2 Forwarding" on page 1 for more details.

42.1.0.1 Routing

Routing is the mechanism of looking up a routing table to determine the best path for a packet from a given sender to reach its destination through intermediate routers. There are two forms of routing, static and dynamic.

- **Static routing:** Suitable for small networks whereby the number of routes is limited and can be manually configured.
- **Dynamic routing:** More suited to larger networks with complex topologies that may change over time. This is an adaptive form of routing which determines the network topology using routing protocols, which are then used to automatically and periodically generate and populate routing tables. Routing tables contain information derived from routing algorithms.

Routing protocols communicate routing information about the connected devices between neighboring routers. The routing table maps an IP address prefix to the next hop address prefix. This is, in essence, a Layer-3 topological view of the network and is optimized for detecting and adapting to changes in the topology.

42.1.0.2 Forwarding

Forwarding is the mechanism of passing or forwarding a packet from one port or interface in the switch to the appropriate egress interface by looking up the forwarding table. Although it is possible to supplement the forwarding table with extra information that is typically found in the routing table, such as next hop information, forwarding statistics, and QoS metrics, it is more common for the routing and forwarding tables to be kept separate. In this way, the routing tables can be used to generate compact and efficient forwarding tables, which are optimized for hardware storage and lookup functionality.

42.1.1 Store and forward

A key function of a store and forward switch is to forward incoming packets to the appropriate destination interface. The mechanism by which packets are forwarded to the appropriate destination interface is called store and forward, whereby the packets received by the switch are stored in an input queue until they reach the head of the queue. Once at the head of the queue, the switch core examines the packets' destination and through a lookup mechanism determines how to forward the packet to its intended destination, forwarding the data through the switch fabric.

Before being forwarded through the switch fabric, the switch core may perform various Layer 2 (MAC layer) and Layer 3 (IP layer) validation checks. See “42.1.1.1 MAC Layer 2 validation” on page 2 and “42.1.1.2 IP Layer 3 validation” on page 2 respectively for more details.

42.1.1.1 MAC Layer 2 validation

- **Ethernet frame validation**: Every Ethernet frame that is forwarded is first validated to ensure it is well-formed, that is, it is within the allowed frame size limits, and that known fields in the Ethernet frame are correct. Layer 3 (IP layer) validation may also occur to ensure the correct IP version field, protocol identifiers etc. are correct, see “42.1.1.2 IP Layer 3 validation” on page 2.
- **Ethernet Frame Check Sequence (FCS) error checking**: The Ethernet MAC FCS is compared against the Cyclic Redundancy Check (CRC) calculated by the store and forward switch. If the Ethernet frames’ FCS differs from the calculated CRC, the frame is considered to contain physical or data link errors and is dropped. In this way, corrupt Ethernet frames are prevented from propagating through the rest of the network.

42.1.1.2 IP Layer 3 validation

- **Packet lifetime control**: Layer 3 switches must also decrement the Time-To-Live (TTL) field in the IP packet header to prevent packets infinitely circulating the network in routing loops. When the TTL value reaches zero, the packet is discarded.
- **Checksum recalculation**: If the Layer 3 switch modifies the TTL, the corresponding IP header checksum and Ethernet FCS need to be recalculated and updated.
- **Fragmentation**: Should the Maximum Transmission Unit (MTU) of the outgoing Ethernet link be smaller than the size of the packet; the packet needs to be fragmented before being forwarded.

If the Ethernet frame is determined to be valid, the switch begins the forwarding process whereby the switch core examines the packets' destination and looks up the forwarding table to determine which egress port (unicast) or ports (multicast/broadcast) are to be used. Since the destination MAC address is the first six bytes of the Ethernet frame, the forwarding process is generally faster than Layer 3 routing table lookup, which requires dissection of the various MAC and IP layer protocol fields.

If there is no entry for a given destination MAC address, the switch does not know where to forward the packet. In this case, the Ethernet frame is forwarded out to all ports on the switch, or flooded. As Ethernet frames are passed through the switch, the switch core updates the MAC forwarding table, noting the source MAC address and the interface on which it arrived. By doing this, the switch core is able to maintain the forwarding table. However, since MAC tables have a finite memory size, entries age out to ensure that the table is up-to-date.

42.2 Xbar switch fabric overview

To realize a flexible store and forward switching solution, a fully interconnected Xbar switching architecture with N input busses and N output busses is implemented where each crosspoint may be either on or off. Curtiss-Wright FTI switches that support this fully interconnected switching architecture are known as Xbar switches. The advantage of the Xbar architecture is its flexibility, enabling complete control over forwarding paths for Ethernet frames using a fully interconnected two-state crosspoint (on or off) switching fabric. Moreover, having high-speed data links in the fabric lowers the switching latency, compared to other switching architectures by minimizing the number of connecting points.

By default, all Ethernet frames received on the input ports are forwarded to all sink devices connected to the output ports. However, using Xbar switching technology it is possible to selectively forward the Ethernet frames from the input ports to specified output ports.

Consider an eight-port switch with four DAUs directly connected on ports 1, 3, 5, and 7, and an unmanaged Ethernet switch on port 6 as shown in the following figure. Ethernet frames transmitted by the DAUs and the switch are forwarded to a number of sink devices, such as a network recorder connected on port 2, an Ethernet-to-PCM gateway connected on port 4 and an analysis laptop connected on port 8.

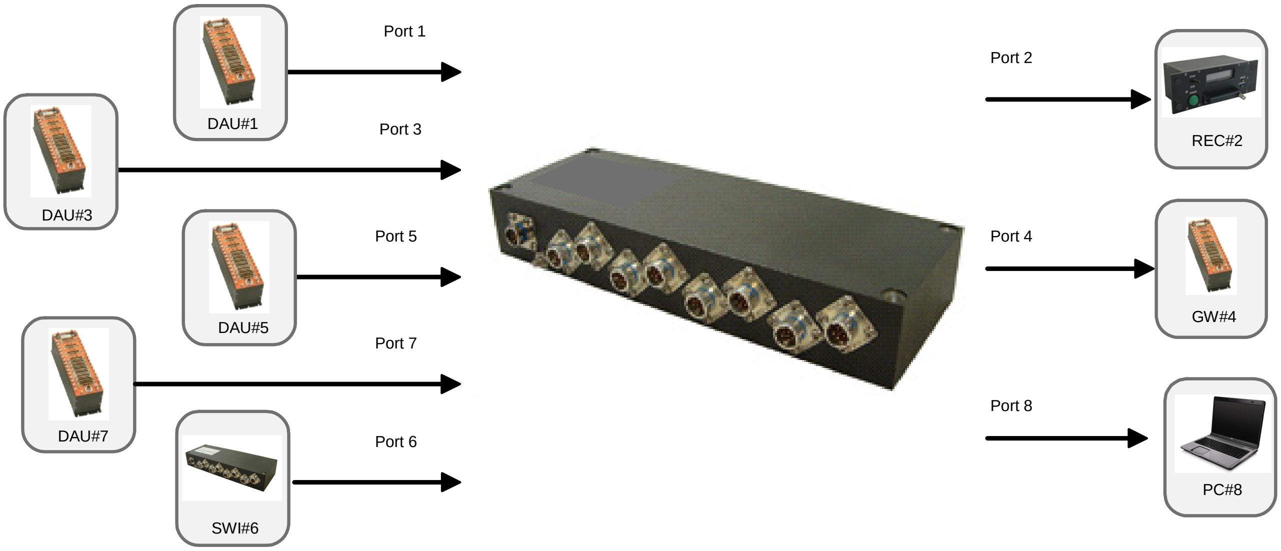


Figure 42-1: Typical FTI switch configuration

The following table summarizes the forwarding paths for each of the network devices connected to the Xbar switch. In this scenario, a DAU (named DAU#1) is connected to port 1 of the Xbar switch. The Ethernet frames transmitted by DAU#1 are received on port 1 and should be forwarded to a number of destination devices: a network recorder (REC#2) connected on port 2; an Ethernet-to-PCM gateway device (GW#4) connected on port 4; and an analysis laptop (named PC#8) connected on port 8.

Table 42-1: Example forwarding configuration

Port number	Input source network node	Destination network node and description
1	DAU#1 Data acquisition unit	• REC#2: All data from this DAU are recorded. • GW#4: A subset of the data streams from this DAU is relayed to the ground for real-time analysis. • PC#8: The DAU is programmed by the PC.
2	REC#2 Network recorder	• GW#4: Transmits its memory utilization to the GW#4 network node where the parameter value is transmitted over the PCM link to be monitored in real-time on the ground. • PC#8: It is programmed from the PC and also can be queried by the PC using SNMP.
3	DAU#3 Data acquisition unit	• REC#2: All data from this DAU are recorded. • PC#8: The DAU is programmed by the PC.
4	GW#4 Ethernet -to-PCM gateway	• None: This device does not transmit any Ethernet packets to any other devices. For example, consider this as an Ethernet bus monitor module that only receives Ethernet frames and is programmed through a KAD/BCU/140 controller that is housed in the same chassis.
5	DAU#5 Data acquisition unit	• GW#4: A subset of the data streams from this DAU is relayed to the ground for real-time analysis. • PC#8: The DAU is programmed by the PC.
6	SWI#6 Network switch	• REC#2: All data streams aggregated through the switch are recorded. • PC#8: The DAU is programmed by the PC.
7	DAU#7 Data acquisition unit	• PC#8: The DAU is programmed by the PC.

Table 42-1: Example forwarding configuration (continued)

Port number	Input source network node	Destination network node and description
8	PC#8 Analysis and programming PC	• All Ports: Programs all devices in the system and analyzes all data from an onboard flying ground station.

The following figure illustrates the Xbar configuration required to realize this forwarding configuration between the DAUs and the various sink devices. The Xbar switching fabric comprises a fully interconnected matrix of crosspoints between the input and output data lines.

Ethernet frames received on the ingress interface of a given port are never forwarded back out on the egress interface of the same port as indicated in the illustration. This blocked data path is indicated by the grey crosspoint, which denotes the interconnection between the ingress and egress interfaces as being blocked. However, Ethernet frames received on a given port can be potentially forwarded to one or more ports. You must explicitly define which interconnections are allowed by enabling the appropriate crosspoint.

The enabled crosspoints are denoted by the hashed crosspoint. All other interconnection paths that have not been enabled are denoted by the clear crosspoint. For example, Ethernet frames received on port 1 from DAU#1 are not forwarded through port 1 back to DAU#1. However, Ethernet frames received on port 1 are to be forwarded to the target devices connected on ports 2, 4 and 8. Similarly, the network recorder, REC#2, connected on port 2 periodically transmits its status information that is to be relayed to the ground via the Ethernet-to-PCM gateway, GW#4, connected on port 4 and to the analysis laptop, PC#8, connected on port 8.

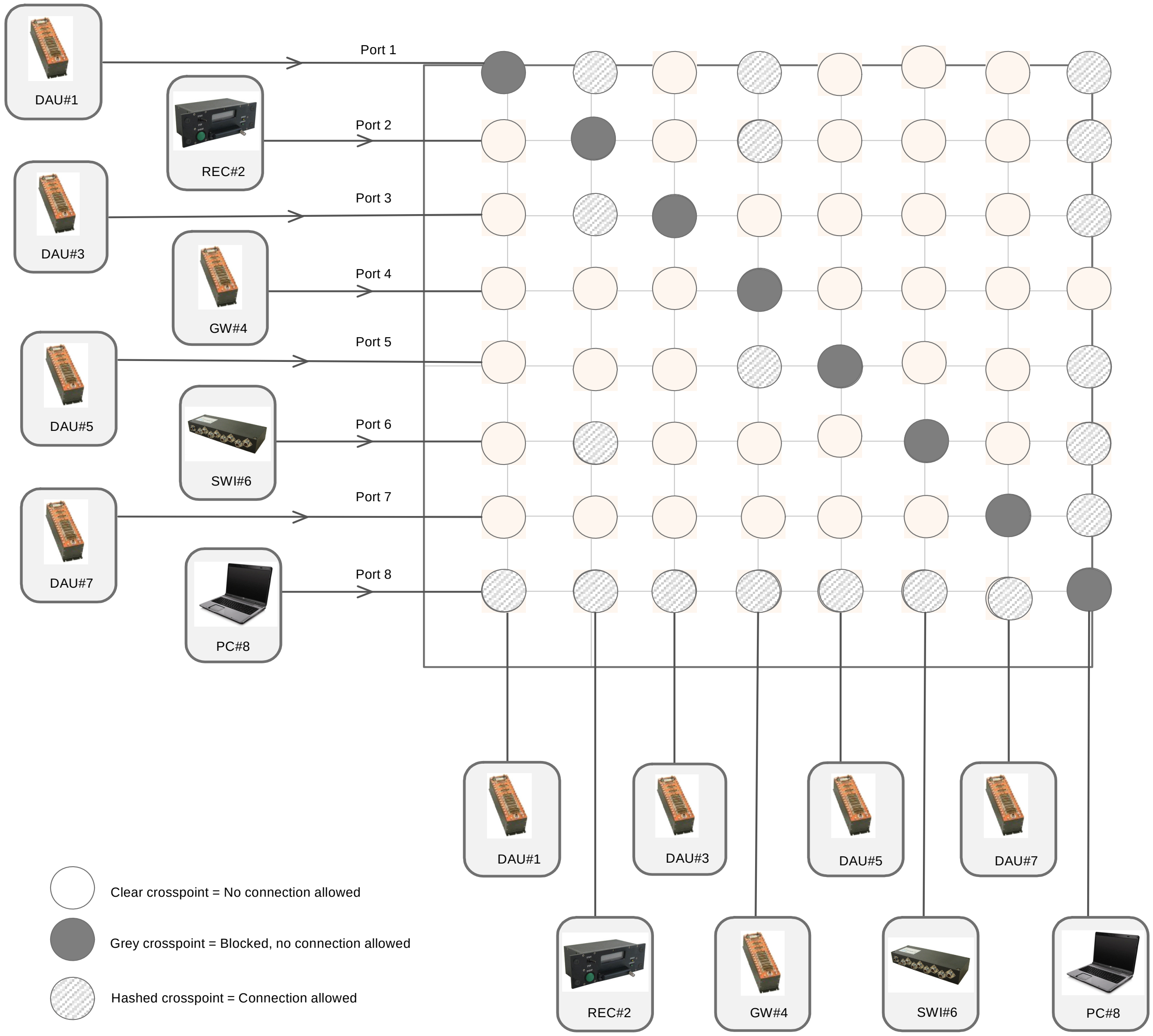

DAU#1
REC#2
DAU#3
GW#4
DAU#5
SWI#6
DAU#7
PC#8
Port 1
Port 2
Port 3
Port 4
Port 5
Port 6
Port 7
Port 8
DAU#1
DAU#3
DAU#5
DAU#7
REC#2
GW#4
SWI#6
PC#8
Clear crosspoint = No connection allowed
Grey crosspoint = Blocked, no connection allowed
Hashed crosspoint = Connection allowed

Figure 42-2: Xbar crosspoint switch fabric

NOTE: A DAU can only be programmed and/or pinged if the link is bi-directional. Therefore the crosspoint between the DAU and programming ports must be enabled. For devices that do not transmit any Ethernet data and only receive Ethernet frames, only those crosspoints which connect the input to output need to be enabled.

The configuration of the crosspoints is used to populate the static forwarding table. The table is considered to be static since it does not rely on adaptive or learning based routing protocols to populate the forwarding table, the table is static and fully-defined by you. The static forwarding table is set up in the Xbar switch using the SNMP. SNMP can be used to modify the static forwarding table at any time without having to interrupt or power cycle the Xbar switch.

42.3 Xbar forwarding and filtering

The previous section described the Xbar forwarding operations. As a consequence of a fully interconnected crosspoint switching fabric, all Ethernet frames received on a given port are forwarded to the appropriate destination if the interconnection is allowed in the forwarding table. However, often finer granularity is required in terms of the forwarding paths whereby only a selected subset of Ethernet frames received from a given input port should be forwarded to a specified output port. To achieve this, a filter is applied to the egress port buffer. The filter is applied only to the egress buffer in order to minimize the processing and lookup required for the forwarding process, thereby minimizing the forwarding latency.

In this section, consider a four-port Xbar switch with a DAU connected on port 1 and port 3 (DAU#1 and DAU#3 respectively) transmitting data to be forwarded to a network recorder (REC#2) and an Ethernet-to-PCM gateway (GW#4) as illustrated in Figure 42-3 on page 7. Defining the Xbar forwarding table alone enables all Ethernet frames transmitted by the DAU to be forwarded to the user-defined allowed destination devices. However, Xbar switching fabric allows for finer granularity with regards to the specification of the forwarding paths, where only Ethernet frames with destination MAC addresses that match the filter criteria may be forwarded through the egress interface of the output port. This mechanism is known as filtering.

For example, DAU#1 is transmitting three packet streams to unique destination multicast addresses:

- **Video data stream** (red): The video stream should only be forwarded to the network recorder, REC#2, but not the Ethernet-to-PCM gateway, GW#4. Essentially, the video stream is filtered from the outgoing egress interface with GW#4.
- **Analog data stream** (green): The analog stream should be forwarded to the Ethernet-to-PCM gateway, GW#4, but not the network recorder, REC#2. Similarly the analog stream is filtered on the output to the network recorder, REC#2.
- **ARINC-429 data stream** (blue): The ARINC-429 is forwarded to both the network recorder, REC#2, and the Ethernet-to-PCM gateway, GW#4.

The complete Xbar forwarding and filtering specification is summarized in the following figure.

Table 42-2: Example forwarding and filtering requirements

Port Number	Input source network node	Destination network node and description
1	DAU#1 Data acquisition unit • Video • Analog • ARINC-429	• Video data stream is forwarded to the network recorder, REC#2, but not the Ethernet-to-PCM gateway., GW#4 • Analog is forwarded to the Ethernet-to-PCM gateway, GW#4, but not the network recorder, REC#2. • ARINC-429 is forwarded to both the network recorder, REC#2, and the Ethernet-to-PCM gateway, GW#4.
2	REC#2 Network recorder • Memory utilization and recorder status	• Transmits its memory utilization to the Ethernet-to-PCM gateway, GW#4, where the parameter value is transmitted over the PCM link to be monitored in real-time on the ground.
3	DAU#3 Data acquisition unit • MIL-STD-1553 • Audio • Temperature	• MIL-STD-1553 is forwarded to the Ethernet-to-PCM gateway, GW#4, but not the network recorder, REC#2. • Audio data stream is forwarded to the network recorder, REC#2, but not the Ethernet-to-PCM gateway, GW#4. • Temperature is forwarded to both the network recorder, REC#2, and the Ethernet-to-PCM gateway, GW#4.
4	GW#4 Ethernet-to-PCM gateway	• None: This device does not transmit any Ethernet packets to any other devices. For example, consider this as an Ethernet bus monitor module that only receives Ethernet frames and is programmed through a KAD/BCU/140 controller that is housed in the same chassis.

There are two steps to set up the Xbar configuration:

1. Define the forwarding table to enable the required interconnection crosspoints.
   In Figure 42-3 on page 7, DAU#1 and DAU#3 must be able to communicate with both REC#2 and GW#4. In addition, REC#2 must be able to communicate with GW#4 in order to forward its memory utilization, health and status. This forwarding configuration is achieved by enabling/disabling the appropriate interconnection crosspoints in the Xbar fabric.
2. Define the filters to be applied to the outgoing ports.
   If the crosspoint is enabled, the forwarding table allows for all Ethernet frames to be forwarded to the specified destination network end nodes. If only a subset of the Ethernet frames are to be forwarded through the egress port, then a filter must be defined and associated with the egress port.

As with the forwarding configuration, SNMP can be used to modify the filter specification at any time without having to interrupt or power cycle the Xbar switch. “42.4.2 Setting the filter type” on page 11 details using SNMP to specify the filtering configuration.

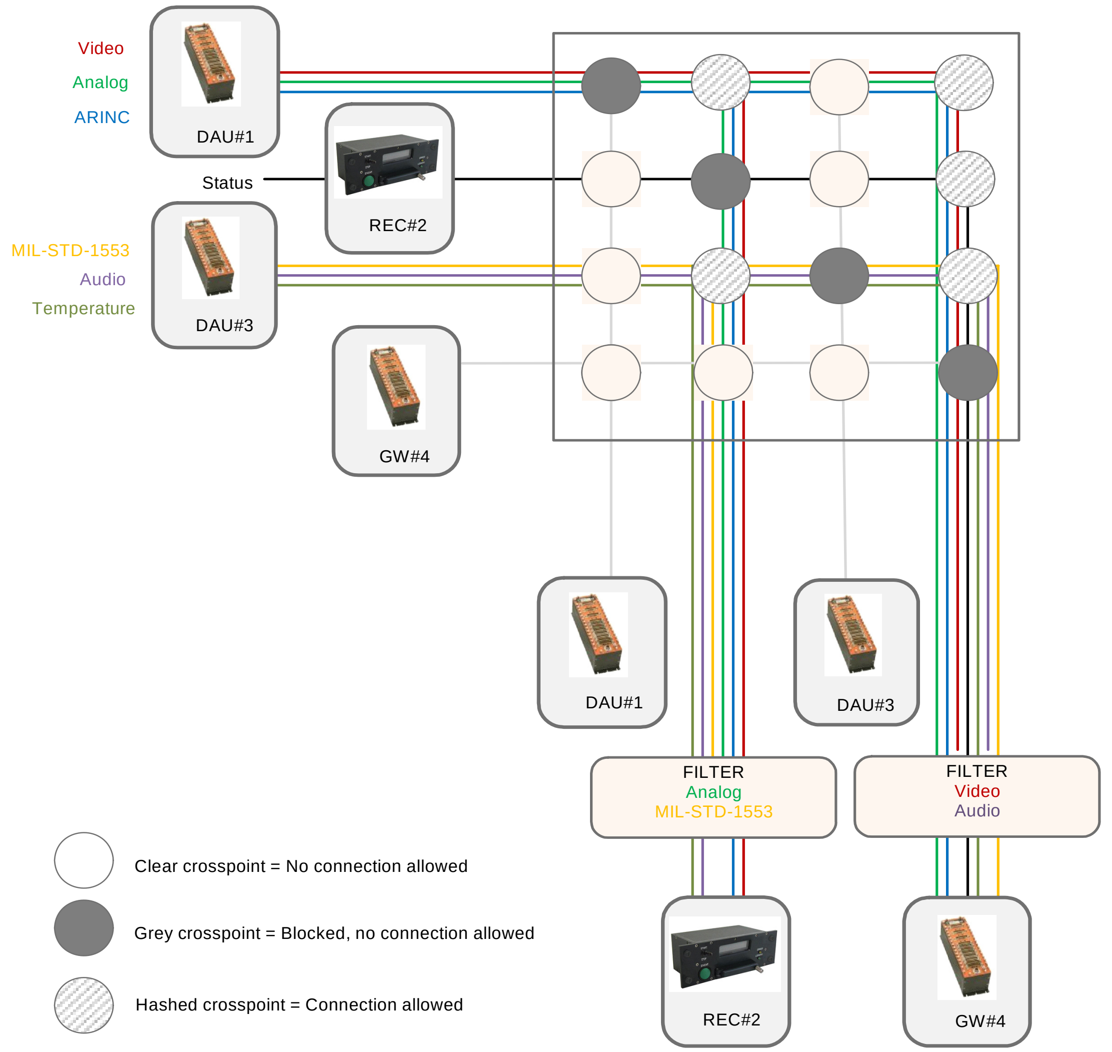


Figure 42-3: Typical forwarding and filtering configuration

42.4 Configuring forwarding and filtering using SNMP

The forwarding and filtering tables are configured in the Xbar switch using the dot1dStatic subtree originally defined in the SNMP Bridge Management Information Base (Bridge-MIB), (IETF RFC 4188).

NOTE: For more information on SNMP, refer to *TEC/NOT/058 - Overview of SNMP and using third party SNMP tools*. The Curtiss-Wright Studio software suite uses SNMP under the hood to interact with the SNMP-enabled devices. However, since SNMP is a standardized technology, for the purposes of illustration the following describes using the Net-SNMP open source utility to configure Xbar switches.

For simplicity, the dot1dStatic subtree in the Bridge-MIB shall be discussed, which is used to configure Xbar forwarding and filtering. The dot1dStatic subtree is a table that comprises a number of entries that define the forwarding and filtering information to be applied to the ports on the Xbar switch. Each entry in the table consists of four variables:

1. **Destination MAC address**: The destination MAC address in an Ethernet frame to which this entry's filtering information applies.
2. **Receive Port Interface**: The port number on which the frame must be received in order for this entry's filtering information to apply.
3. **Bitvector of the Allowed Outgoing Ports**: The set of ports to which this Ethernet frame is allowed to be forwarded. The bit vector is used to represent the on/off state or forwarding for each port in the switch.
4. **Entry Persistence**: States the persistence of this forwarding/filtering entry indicating if it is permanent until removed, deleted on reset or deleted on timeout.

42.4.1 Setting the forwarding table

The first N entries in the table are used to define the allowed forwarding paths in the Xbar switch, where N is the number of ports on the switch, that is, if the switch has eight-ports then the first eight entries of the table are used to define the forwarding configuration.

To define a forwarding entry the Receive Port Interface, Bitvector of the Allowed Outgoing Ports and Entry Persistence are used, see the following table**.**

NOTE: Port 1 is represented by the Least Significant Bit (LSB) of the bit vector. Any attempt to set other values in these fields using SNMP results in a "Bad Value" error message using an SNMP tool.

Table 42-3: Tabular representation of the dot1dStatic MIB

Receive Port Interface	Destination MAC address	Bitvector of the Allowed Outgoing Ports	Entry Persistance
1	Used to define filter types, described in “42.4.2 Setting the filter type” on page 11.		Permanent
2			Permanent
3			Permanent
4			Permanent
5			Permanent
6			Permanent
7			Permanent
8			Permanent

The generic form for the SNMP command used to specify the forwarding configuration is as follows:

```
snmpset [Version] [Community][Agent] [OID] [Type] [Value]
```

Thus to specify the forwarding for a given port N, the SNMP command arguments are:

- `[Version]:` `-v2c`
- `[Community]:` `-c public`
- `[Agent]:` `192.168.1.1` which is the IP address of the Xbar switch being configured.
- `[OID]:` `.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTable.dot1dStaticEntry.dot1dStaticAllowedToGoTo.N,` where N is the port number.
- `[Type]:` `x` where x indicates that the following value is a hex value.
- `[Value]:` `0a` where 0x0a is the bit vector of Allowed outgoing ports with port 1 being the LSB of the vector.

The complete snmpset command for the forwarding configuration of port 1 is:

```
snmpset.exe-v2c-cpublic192.168.1.1.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTable.dot1dStaticEntry.dot1dStaticAllowedToGoTo.1 x 0a
```

For more information on using SNMP see *TEC/NOT/058 - Overview of SNMP and using third party SNMP tools.*

42.4.1.1 Example forwarding configuration

Consider the following example of an eight-port Xbar switch with only four devices connected to it, see the following figure.

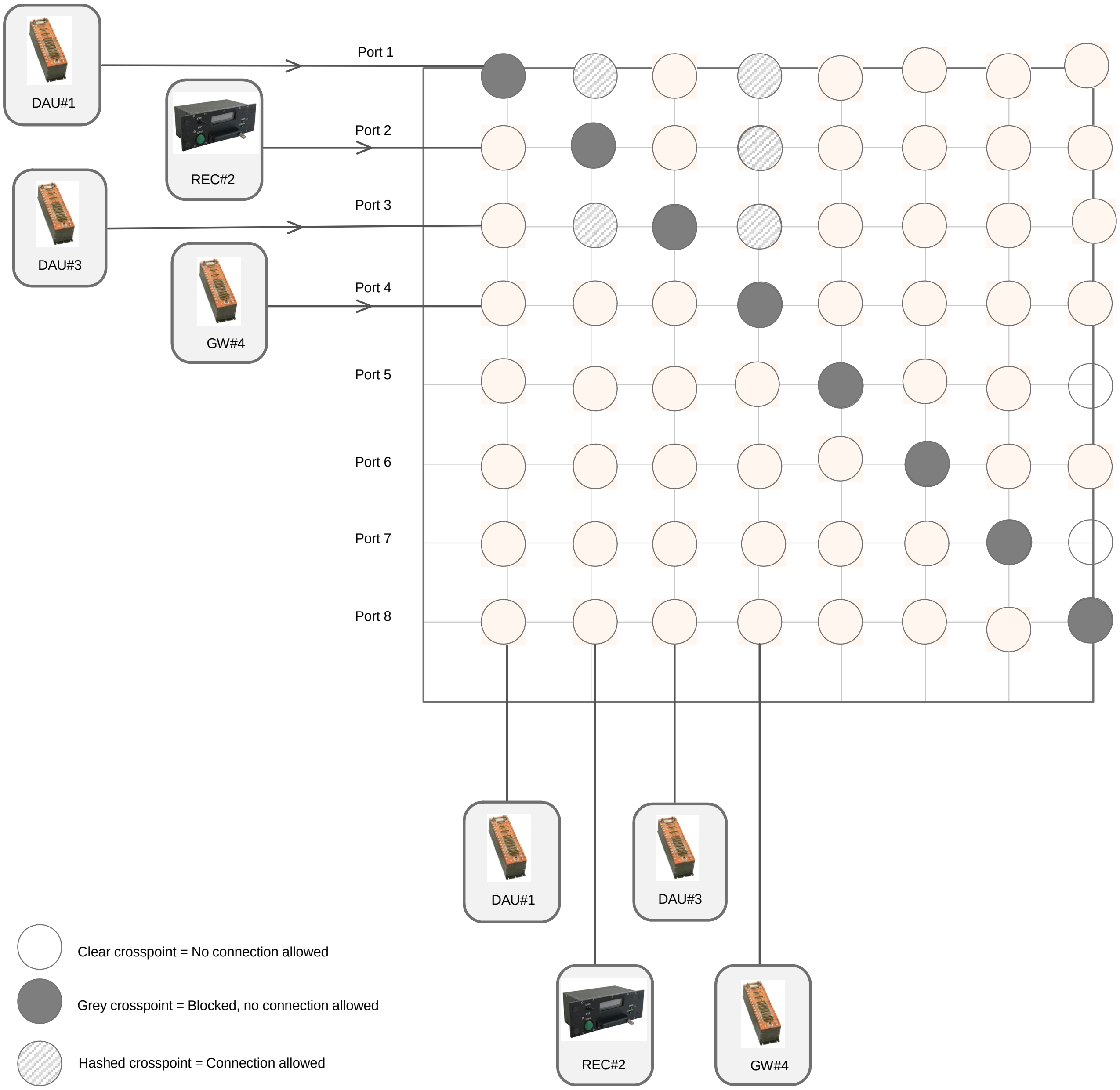

DAU#1
REC#2
DAU#3
GW#4
Port 1
Port 2
Port 3
Port 4
Port 5
Port 6
Port 7
Port 8
DAU#1
DAU#3
REC#2
GW#4
Clear crosspoint = No connection allowed
Grey crosspoint = Blocked, no connection allowed
Hashed crosspoint = Connection allowed

Figure 42-4: Eight-port Xbar forwarding configuration

See the following table for the SNMP commands for the forwarding configuration of each port in Figure 42-3 on page 7.

Table 42-4: Description of the Xbar forwarding SNMP command syntax

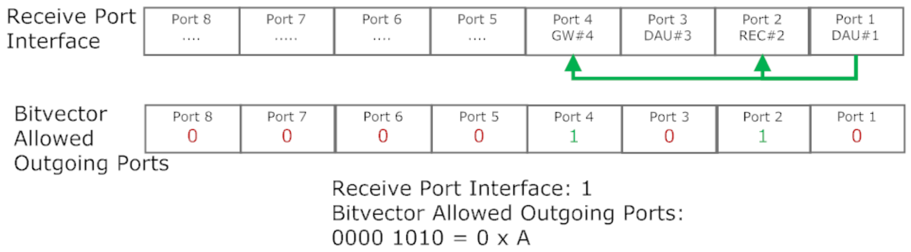
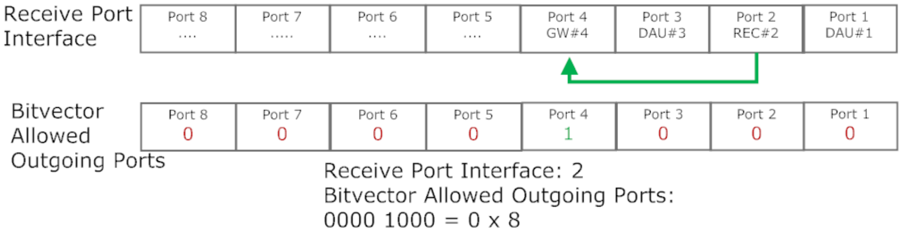
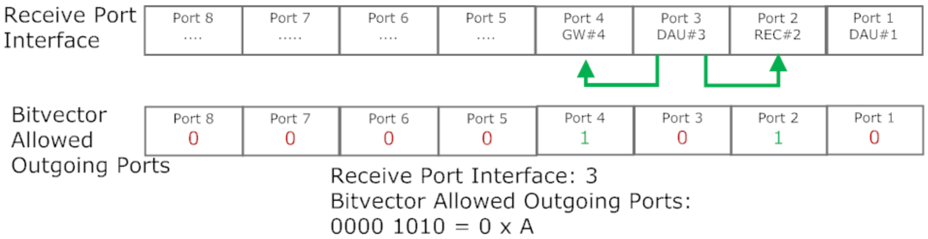
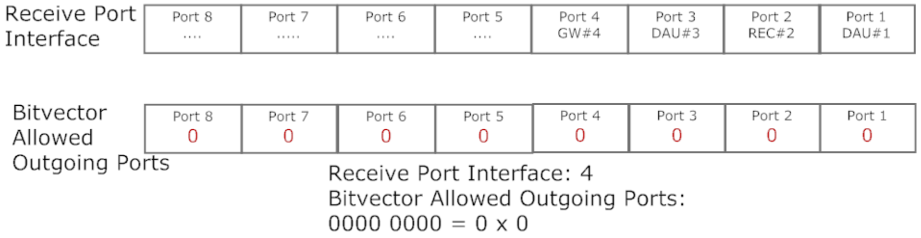
Receive Port Interface	Bitvector of the Allowed Outgoing Ports
Port 1 DAU#1	 Receive Port Interface Port 8 .... Port 7 ..... Port 6 .... Port 5 .... Port 4 GW#4 Port 3 DAU#3 Port 2 REC#2 Port 1 DAU#1 Bitvector Allowed Outgoing Ports Port 8 0 Port 7 0 Port 6 0 Port 5 0 Port 4 1 Port 3 0 Port 2 1 Port 1 0 Receive Port Interface: 1 Bitvector Allowed Outgoing Ports: 0000 1010 = 0 x A
	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTable.dot1dStaticEntry.dot1dStaticAllowedToGoTo.1 x 0a
Port 2 REC#2	 Receive Port Interface Port 8 .... Port 7 ..... Port 6 .... Port 5 .... Port 4 GW#4 Port 3 DAU#3 Port 2 REC#2 Port 1 DAU#1 Bitvector Allowed Outgoing Ports Port 8 0 Port 7 0 Port 6 0 Port 5 0 Port 4 1 Port 3 0 Port 2 0 Port 1 0 Receive Port Interface: 2 Bitvector Allowed Outgoing Ports: 0000 1000 = 0 x 8
	snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTable.dot1dStaticEntry.dot1dStaticAllowedToGoTo.2 x 08
Port 3 DAU#3	 Receive Port Interface Port 8 .... Port 7 ..... Port 6 .... Port 5 .... Port 4 GW#4 Port 3 DAU#3 Port 2 REC#2 Port 1 DAU#1 Bitvector Allowed Outgoing Ports Port 8 0 Port 7 0 Port 6 0 Port 5 0 Port 4 1 Port 3 0 Port 2 1 Port 1 0 Receive Port Interface: 3 Bitvector Allowed Outgoing Ports: 0000 1010 = 0 x A
	snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTable.dot1dStaticEntry.dot1dStaticAllowedToGoTo.3 x 0a
Port 4 GW#4	 Receive Port Interface Port 8 .... Port 7 ..... Port 6 .... Port 5 .... Port 4 GW#4 Port 3 DAU#3 Port 2 REC#2 Port 1 DAU#1 Bitvector Allowed Outgoing Ports Port 8 0 Port 7 0 Port 6 0 Port 5 0 Port 4 0 Port 3 0 Port 2 0 Port 1 0 Receive Port Interface: 4 Bitvector Allowed Outgoing Ports: 0000 0000 = 0 x 0
	snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTable.dot1dStaticEntry.dot1dStaticAllowedToGoTo.4 x 00

Table 42-4: Description of the Xbar forwarding SNMP command syntax (continued)

Ports 5 - 8
No devices connected
	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTable.dot1dStaticEntry.dot1dStaticAllowedToGoTo.5 x 00`
	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTable.dot1dStaticEntry.dot1dStaticAllowedToGoTo.6 x 00`
	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTable.dot1dStaticEntry.dot1dStaticAllowedToGoTo.7 x 00`
	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTable.dot1dStaticEntry.dot1dStaticAllowedToGoTo.8 x 00`

The resulting forwarding table appears as follows:

Table 42-5: Resulting forwarding dot1dStatic MIB table entries

Receive Port Interface	Destination MAC address	Bitvector of the Allowed Outgoing Ports	Entry Persistance
1	Used to define filter types, described in “42.4.2 Setting the filter type” on page 11.	0x000A	Permanent
2		0x0008	Permanent
3		0x000A	Permanent
4		0x0000	Permanent
5		0x0000	Permanent
6		0x0000	Permanent
7		0x0000	Permanent
8		0x0000	Permanent

42.4.2 Setting the filter type

Xbar switches have filtering capabilities, which may be configured to pass or reject unicast, multicast, or broadcast packets being forwarded out of each port.

The following three types of filter can be specified:

- Filter unicast
- Filter broadcast
- Filter multicast

Filter unicast

The unicast filter is described by the SNMP OID variable contained in the Acra MIB. More details on the variable structure can be found in “42.5 Appendix: Xbar connection SNMP variables” on page 20.

This variable specifies whether unicast traffic is allowed at this output. The `FilterUnicast` variable may be set to the following values:

- `Allowed (0)`: All unicast packets are allowed out of this connection.
- `Blocked (1)`: No unicast packets are allowed out of this connection.

This does not affect unicast packets arriving at this connection. Such packets can be forwarded to other outputs, subject to routing defined in the dot1dStaticAddress array, regardless of this setting.

NOTE: The unicast filter and broadcast filter can only be Blocked or Allowed. The PassFilter and RejectFilter options can only be used with a multicast filter. If an attempt is made to apply a PassFilter or RejectFilter setting value to either a unicast filter or a broadcast filter, an "Unsupported" error message is returned.

When filtering, it is important to understand that if a unicast filter is applied to a given port, it is not possible to use SNMP, ping, or program the device connected on this port. As a precautionary measure, by default, the last port or port N on a N-port Xbar switch can never be set to block unicast in order to prevent permanently blocking access to the switch.

To specify the unicast filter for a given port N, the snmpset command arguments are:

- `[Version]:` `-v2c`
- `[Community]:` `-c public`
- `[Agent]:` `192.168.1.1` which is the IP address of the Xbar switch being configured.
- `[OID]:` `.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.N`, where N is the port number.
- `[Type]:` `i` where i indicates that the following value is a integer value.
- `[Value]:` where the possible values for this variable are: {Allowed(0), Blocked(1)}

Therefore, the complete snmpset command to set the unicast filter configuration of port 1 to '`Allowed`' is:

```
snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.
connection.connectionTable.connectionEntry.connectionFilterUnicast.1 i 0
```

For more information on using SNMP see *TEC/NOT/058 - Overview of SNMP and using third party SNMP tools.*

Filter unicast examples

The following two tables provide examples of setting the unicast filter for an eight-port Xbar switch.

Table 42-6: Filter unicast configuration example 1

Port number	1	2	3	4	5	6	7	8
Unicast traffic	Allow	Block	Allow	Block	Allow	Block	Allow	Allow

Port 1	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.1 i 0`
Port 2	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.2 i 1`
Port 3	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.3 i 0`
Port 4	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.4 i 1`
Port 5	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.5 i 0`
Port 6	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.6 i 1`
Port 7	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.7 i 0`

Table 42-6: Filter unicast configuration example 1 (continued)

Port 8	snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.8 i 0

Table 42-7: Filter unicast configuration example 2

Port number	1	2	3	4	5	6	7	8
Unicast traffic	Allow	Allow	Allow	Allow	Block	Block	Block	Allow

Port 1	snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.1 i 0
Port 2	ssnmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.2 i 0
Port 3	snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.3 i 0
Port 4	snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.4 i 0
Port 5	snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.5 i 1
Port 6	snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.6 i 1
Port 7	snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.7 i 1
Port 8	snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast.8 i 0

Filter broadcast

The broadcast filter is similar to the unicast filter type. More details on the variable structure can be found in “42.5 Appendix: Xbar connection SNMP variables” on page 20. This variable specifies whether broadcast traffic is allowed at this output. The `FilterBroadcast` variable may be set to the following values:

- `Allowed (0)`: All broadcast packets are allowed out of this connection.
- `Blocked (1)`: No broadcast packets are allowed out of this connection.

This does not affect broadcast packets arriving at this connection. Such packets can be forwarded to other outputs, subject to routing defined in the dot1dStaticAddress array, regardless of this setting**.**

NOTE: The unicast filter and broadcast filter can only be Blocked or Allowed. The PassFilter and RejectFilter options can only be used with a multicast filter. If an attempt is made to apply a PassFilter or RejectFilter setting value to either a unicast filter or broadcast filter an "Unsupported" error message is returned.

When filtering, it is important to understand that if a broadcast filter is applied to a given port, it is not possible to use SNMP, ping, or program the device connected on this port. As a precautionary measure, by default, the last port or port

N on a N-port Xbar switch can never be set to block broadcast in order to prevent permanently blocking access to the switch.

Filter broadcast examples

The following two tables provide examples of setting the broadcast filter for an eight-port Xbar switch.

Table 42-8: Filter broadcast configuration example 1

Port number	1	2	3	4	5	6	7	8
Broadcast traffic	Allow	Block	Allow	Block	Allow	Block	Allow	Allow

Port 1	`snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.1 i 0`
Port 2	`ssnmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.2 i 1`
Port 3	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.3 i 0`
Port 4	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.4 i 1`
Port 5	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.5 i 0`
Port 6	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.6 i 1`
Port 7	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.7 i 0`
Port 8	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.8 i 0`

Table 42-9: Filter broadcast configuration example 2

Port number	1	2	3	4	5	6	7	8
Broadcast traffic	Allow	Allow	Allow	Allow	Block	Block	Block	Allow

Port 1	`snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.1 i 0`
Port 2	`ssnmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.2 i 0`
Port 3	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.3 i 0`

Table 42-9: Filter broadcast configuration example 2 (continued)

Port 4	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.4 i 0`
Port 5	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.5 i 1`
Port 6	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.6 i 1`
Port 7	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.7 i 1`
Port 8	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast.8 i 0`

Filter multicast

Xbar switches support multicast filtering on the output. This variable specifies whether multicast traffic is allowed at this output. The `FilterMulticast` variable may be set to the following values:

- `Allowed (0)`: All multicast packets are allowed out of this connection.
- `Blocked (1)`: No multicast packets are allowed out of this connection.
- `PassFilter (2)`: The onlymulticast packets allowed out of this connection are those whose destination MAC is in the dot1dStatic table, with a 1 for this connection in the `AllowedToGoTo` value.
- `RejectFilter (3)`: All multicast packets are allowed out of this connection EXCEPT those whose destination MAC is in the dot1dStatic table, with a 1 for this connection in the `AllowedToGoTo` value.

This variable indicates the multicast filter type associated with a given output port. However, the dot1dStatic subtree in the Bridge-MIB is also used to enter the specific multicast addresses to which this filter is applied. Xbar switches can store up to 32 entries in the dot1dStatic table of which the first N entries of the table are used to configure the forwarding configuration of an N-port switch as described earlier. The remaining entries are available to set the specific multicast addresses used for filtering.

As before, the filter type associated with each port is configured.

NOTE: While Xbar switches can store up to 32 entries, the 32nd entry in the dot1dStatic table is reserved for PTP traffic and cannot be edited.

Filter multicast examples

The following two tables provide examples of setting the multicast filter for an eight-port Xbar switch.

Table 42-10: Filter multicast configuration example 1

Port number	1	2	3	4	5	6	7	8
Multicast traffic	Reject	Pass	Allow	Reject	Pass	Allow	Reject	Pass

Port 1	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.1 i 3`
Port 2	`snmpset.exe -v2c -c public 192.168.1.1.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.2 i 2`

Table 42-10: Filter multicast configuration example 1 (continued)

Port 3	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.3 i 0
Port 4	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.4 i 3
Port 5	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.5 i 2
Port 6	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.6 i 0
Port 7	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.7 i 3
Port 8	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.8 i 2

Table 42-11: Filter multicast configuration example 2

Port number	1	2	3	4	5	6	7	8
Multicast traffic	Allow	Pass	Reject	Allow	Pass	Reject	Allow	Pass

Port 1	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.1 i 0
Port 2	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.2 i 2
Port 3	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.3 i 3
Port 4	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.4 i 0
Port 5	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.5 i 2
Port 6	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.6 i 3
Port 7	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.7 i 0
Port 8	snmpset.exe -v2c -c public 192.168.1.1 .iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterMulticast.8 i 2

Specifying the multicast addresses to filter

To define a multicast filter entry the Destination MAC Address and Bitvector of Allowed Outgoing Ports are used. The general form for the table to store the multicast address filter information used in the dot1dStatic subtree is shown in the following table.

Table 42-12: Tabular representation of the dot1dStatic MIB

Receive Port Interface	Destination MAC address	Bitvector of the Allowed Outgoing Ports	Entry Persistance
Used to define the forwarding configuration only, described in “42.4.1 Setting the forwarding table” on page 8. In the context of filtering it is used as an index of the filter entries.			Permanent
			Permanent
			Permanent
			Permanent
			Permanent
			Permanent
			Permanent
			Permanent

NOTE: For more information on the notation used to specify the Bitvector of the Allowed Outgoing Ports refer to “42.4.1 Setting the forwarding table” on page 8 that describes the forwarding configuration.

The following table assumes an eight-port Ethernet switch. These filtering entries are for illustrative purposes only. It is assumed that the forwarding configuration has already been defined and the appropriate crosspoints have been enabled between the inputs and outputs. Since the first eight entries are used to specify the forwarding configuration, subsequent entries at index 9 are used to specify the multicast address used for filtering.

Table 42-13: Setting the multicast address filter

<table>
<tr><th>Table entry index</th><th>Description</th></tr>
<tr><td>Index entry 9</td><td>
<table>
<tr><th></th><th>Destination IP</th><th>Destination MAC</th><th>Port 1</th><th>Port 2</th><th>Port 3</th><th>Port 4</th><th>Port5</th><th>Port 6</th><th>Port 7</th><th>Port 8</th></tr>
<tr><td><b>Index 9</b></td><td>224.0.1.1</td><td>01005E000001</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
Set the multicast address upon which to filter for the 9th entry in the table.
<pre>
snmpset.exe -v2c -c public 192.168.1.1
.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTabl
e.dot1dStaticEntry.dot1dStaticAddress.9 x 01005E000001
</pre>
This multicast address is <code>AllowedToGoTo</code> the specified ports.
<pre>
snmpset.exe -v2c -c public 192.168.1.1
.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTabl
e.dot1dStaticEntry.dot1dStaticAllowedToGoTo.9 x 01
</pre>
If port 1 has a multicast filter setting:
<ul>
<li><b>Allowed</b>: All Ethernet frames including those with this multicast address are forwarded to port 1 and allowed to pass through.</li>
<li><b>Blocked</b>: All Ethernet frames including those with this multicast address are forwarded to port 1 but are blocked and do not pass through.</li>
<li><b>PassFilter</b>: Ethernet frames with this multicast address are forwarded to port 1 and allowed to pass through.</li>
<li><b>RejectFilter</b>: Ethernet frames with this multicast address are forwarded to port 1 but are rejected and do not pass through the filter.</li>
</ul>
If port 2 has a multicast filter setting
<ul>
<li><b>Allowed</b>: All Ethernet frames including those with this multicast address are forwarded to port 2 and allowed to pass through.</li>
<li><b>Blocked</b>: All Ethernet frames including those with this multicast address are forwarded to port 2 but are blocked and do not pass through.</li>
<li><b>PassFilter</b>: Ethernet frames with this multicast address are forwarded to port 2.</li>
<li><b>RejectFilter</b>: Ethernet frames with this multicast address are forwarded to port 2.</li>
</ul>
This same interpretation can be applied to each of the ports on the switch.
</td></tr>
<tr><td>Index entry 10</td><td>
<table>
<tr><th></th><th>Destination IP</th><th>Destination MAC</th><th>Port 1</th><th>Port 2</th><th>Port 3</th><th>Port 4</th><th>Port5</th><th>Port 6</th><th>Port 7</th><th>Port 8</th></tr>
<tr><td><b>Index 10</b></td><td>224.0.1.2</td><td>01005E000002</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<pre>
snmpset.exe -v2c -c public 192.168.1.1
.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTabl
e.dot1dStaticEntry.dot1dStaticAddress.10 x 01005E000001

snmpset.exe -v2c -c public 192.168.1.1
.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTabl
e.dot1dStaticEntry.dot1dStaticAllowedToGoTo.10 x 02
</pre>
</td></tr>
</table>

Table 42-13: Setting the multicast address filter (continued)

Index entry 11

	Destination IP	Destination MAC	Port 1	Port 2	Port 3	Port 4	Port5	Port 6	Port 7	Port 8
Index 11	224.0.1.3	01005E000003	0	0	1	0	0	0	0	0

```
snmpset.exe -v2c -c public 192.168.1.1
.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTabl
e.dot1dStaticEntry.dot1dStaticAddress.11 x 01005E000003

snmpset.exe -v2c -c public 192.168.1.1
.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTabl
e.dot1dStaticEntry.dot1dStaticAllowedToGoTo.11 x 04
```

Index entry 12

	Destination IP	Destination MAC	Port 1	Port 2	Port 3	Port 4	Port5	Port 6	Port 7	Port 8
Index 12	224.0.1.4	01005E000004	0	0	0	1	0	0	0	0

```
snmpset.exe -v2c -c public 192.168.1.1
.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTabl
e.dot1dStaticEntry.dot1dStaticAddress.12 x 01005E000004

snmpset.exe -v2c -c public 192.168.1.1
.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTabl
e.dot1dStaticEntry.dot1dStaticAllowedToGoTo.12 x 08
```

Index entry 13

	Destination IP	Destination MAC	Port 1	Port 2	Port 3	Port 4	Port5	Port 6	Port 7	Port 8
Index 13	224.0.1.5	01005E000005	0	0	0	0	1	0	0	0

```
snmpset.exe -v2c -c public 192.168.1.1
.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTabl
e.dot1dStaticEntry.dot1dStaticAddress.13 x 01005E000005

snmpset.exe -v2c -c public 192.168.1.1
.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTabl
e.dot1dStaticEntry.dot1dStaticAllowedToGoTo.13 x 10
```

Resetting filter multicast entries

To clear the multicast filter associated with a given port, the `dot1dStaticStatus` variable in the dot1dStatic is used.

OID: `1.3.6.1.2.1.17.5.1.1.4`

Path: `iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTable.dot1dStaticEntry.dot1dStaticStatus`

The `dot1dStaticStatus` variable may be set to the following values:

- `Other (1)`: This entry is currently in use but the conditions under which it remains so are different from each of the following values.
- `Invalid (2)`: Writing this value to the object removes the corresponding entry.
- `Permanent (3)`: This entry is currently in use and remains so after the next reset of the bridge.
- `deleteOnReset (4)`: This entry is currently in use and remains so until the next reset of the bridge.
- `deleteOnTimeout (5)`: This entry is currently in use and remains so until it is aged out.

This object indicates the status of this entry. The default value is `permanent (3)`.

To remove the Nth entry in the table:

1. Set the `AllowedToGoTo` variable to `0xFF`.

```
snmpset.exe-v2c-cpublic192.168.1.1.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTable.dot1dStaticEntry.dot1dStaticAllowedToGoTo.N x FF
```

2. Change the status of the entry to `invalid.`
   This effectively removes the entry.

```
snmpset.exe-v2c-cpublic192.168.1.1.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dStatic.dot1dStaticTable.dot1dStaticEntry.dot1dStaticStatus.N i 2
```

42.5 Appendix: Xbar connection SNMP variables

The connection settings of the Ethernet ports on the Xbar switch are accessed and modified using the connectionTable subtree in the Acra MIB, see the following table.

The full OID path to the connectionTable is given as:

iso	org	dod	internet	private	enterprises	acra	connection	connection Table
1	3	6	1	4	1	33698	1	14

The format of this table structure comprises a sequence of connectionEntries where each entry is identified by an index and has a number of settings such as connectionSpeed and connectionFilter settings associated with it as shown in the following table.

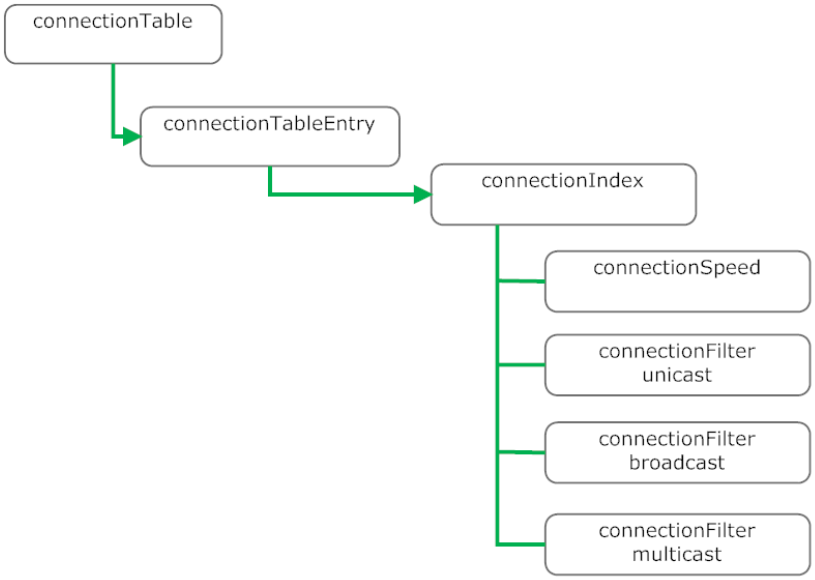


Figure 42-5: connectionTable subtree

42.5.0.1 Connection speed

The speed at which each port operates can be specified using the `connectionSpeed` variable. The allowed values for the connection speed are {10, 100, 1000, Auto} in megabits per second (Mbps). The speed is described by the SNMP OID variable contained in the Acra MIB:

OID:`.1.3.6.1.4.1.33698.14.1.1.2`

Path:`.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionSpeed`

iso	org	dod	internet	private	enterprises	acra	connection	connection Table	connection Entry	connection Speed
1	3	6	1	4	1	33698	14	1	1	2

42.5.0.2 Unicast filter

The unicast filter is described by the SNMP OID variable contained in the Acra MIB:

OID:.1.3.6.1.4.1.33698.14.1.1.3

Path:.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterUnicast

iso	org	dod	internet	private	enterprises	acra	connection	connectionTable	connectionEntry	connectionFilterUnicast
1	3	6	1	4	1	33698	14	1	1	3

42.5.0.3 Broadcast filter

The broadcast filter is described by the SNMP OID variable contained in the Acra MIB:

OID:.1.3.6.1.4.1.33698.14.1.1.4

Path:.iso.org.dod.internet.private.enterprises.acra.connection.connectionTable.connectionEntry.connectionFilterBroadcast

iso	org	dod	internet	private	enterprises	acra	connection	connectionTable	connectionEntry	connectionFilterBroadcast
1	3	6	1	4	1	33698	14	1	1	4

42.5.1 Connection table MIB listing

The Acra MIB detail for the connectionTable is listed below.
For more details on how to read the SNMP MIB see *TEC/NOT/058 - Overview of SNMP and using third party SNMP tools.*

```
connectionTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ConnectionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Control each ethernet connection to the module"
        ::= { connection 1 }

connectionEntry OBJECT-TYPE
    SYNTAX      ConnectionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Settings for one ethernet connection on the module"
    INDEX { connectionIndex }
        ::= { connectionTable 1 }

ConnectionEntry ::= SEQUENCE {
        connectionIndex
            INTEGER,
        connectionSpeed
            Integer32,
        connectionFilterUnicast
            INTEGER,
        connectionFilterBroadcast
            INTEGER,
        connectionFilterMulticast
            INTEGER
        }

connectionIndex OBJECT-TYPE
    SYNTAX      INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Index of the connection"
        ::= { connectionEntry 1 }

connectionSpeed OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Set the interface speed. 0=Autonegotiate, 10=10BaseT,
         100=100BaseTX, 1000=Gigabit"
        ::= { connectionEntry 2 }

connectionFilterUnicast OBJECT-TYPE
    SYNTAX      INTEGER {
                          Allowed (0),
                          Blocked (1),
                          PassFilter (2),
                          RejectFilter (3)
                        }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "Specifies whether unicast traffic is allowed at this output.
       Allowed (0) means all unicast packets are allowed out of this
           connection.
       Blocked (1) means that no unicast packets are allowed out of this
           connection.
       PassFilter (2) means that the only unicast packets allowed out of
           this connection are those whose destination MAC is in the
           dot1dStatic table, with a '1' for this connection in the
           AllowedToGoTo value.
       RejectFilter (3) means that all unicast packets are allowed out of
           this connection EXCEPT those whose destination MAC is in the
           dot1dStatic table, with a '1' for this connection in the
           AllowedToGoTo value.
       This does not affect unicast packets arriving at this connection. Such
       packets can be forwarded to other outputs, subject to routing defined
       in the dot1dStaticAddress array, regardless of this setting."
    ::= { connectionEntry 3 }

connectionFilterBroadcast OBJECT-TYPE
    SYNTAX      INTEGER {
                          Allowed (0),
                          Blocked (1),
                          PassFilter (2),
                          RejectFilter (3)
                        }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "Specifies whether broadcast traffic is allowed at this output.
       Allowed (0) means all broadcast packets are allowed out of this
           connection.
       Blocked (1) means that no broadcast packets are allowed out of this
           connection.
       PassFilter (2) means that the only broadcast packets allowed out of
           this connection are those whose destination MAC is in the
           dot1dStatic table, with a '1' for this connection in the
           AllowedToGoTo value.
       RejectFilter (3) means that all broadcast packets are allowed out of
           this connection EXCEPT those whose destination MAC is in the
           dot1dStatic table, with a '1' for this connection in the
           AllowedToGoTo value.
       This does not affect broadcast packets arriving at this connection.
       Such packets can be forwarded to other outputs, subject to routing
       defined in the dot1dStaticAddress array, regardless of this setting."
    ::= { connectionEntry 4 }

connectionFilterMulticast OBJECT-TYPE
    SYNTAX      INTEGER {
                          Allowed (0),
                          Blocked (1),
                          PassFilter (2),
                          RejectFilter (3)
                        }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "Specifies whether multicast traffic is allowed at this output.
       Allowed (0) means all multicast packets are allowed out of this
           connection.
       Blocked (1) means that no multicast packets are allowed out of this
           connection.
       PassFilter (2) means that the only multicast packets allowed out of
           this connection are those whose destination MAC is in the
           dot1dStatic table, with a '1' for this connection in the
           AllowedToGoTo value.
       RejectFilter (3) means that all multicast packets are allowed out of
           this connection EXCEPT those whose destination MAC is in the
           dot1dStatic table, with a '1' for this connection in the
           AllowedToGoTo value.
       This does not affect multicast packets arriving at this connection.
       Such packets can be forwarded to other outputs, subject to routing
       defined in the dot1dStaticAddress array, regardless of this setting."
        ::= { connectionEntry 5 }
```

42.6 Recommended reading

To better understand this paper, read the following documents.

Table 42-14: Data sheets

Document	Description
NET/SWI/003	8-port Gigabit Ethernet switch with configurable static forwarding and filtering.
NET/SWI/005	16-port Gigabit Ethernet switch with configurable static forwarding and filtering.

Table 42-15: Technical notes

Document	Description
TEC/NOT/058	*Overview of SNMP and using third party SNMP tools*.