DIGITAL GIGAswitch/FDDI System

Release Notes Version 3.1

Part Number: AA-PZT9F-TE

GIGAswitch/FDDI Firmware Baselevels:

- Switch Control Processor (SCP) OP 3.10, BB 3.10, DL 1.00
- Two-port FDDI GIGAswitch Linecard (FGL-2) 3.10
- Four-port FDDI GIGAswitch Linecard (FGL-4) 3.10
- Two-port ATM GIGAswitch Linecard (AGL-2) 3.10
- Two-port ATM GIGAswitch Linecard (AGL-2+) 3.10
- Clock Card (CLK) 3.00
- Power System Controller (PSC) 2.00
- Management Information Base (MIB) 3.10

Digital Equipment Corporation
Maynard, Massachusetts

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from Digital or an authorized sublicensor.

Contents

Preface

Preface

Purpose of This Document

This document describes new features, documentation changes, bug fixes, problems, and restrictions that pertain to the GIGAswitch/FDDI V3.1 firmware release.

Intended Audience

The GIGAswitch/FDDI System Release Notes are intended for customers and Digital Service personnel. Read the release notes before you install, service, or use the GIGAswitch/FDDI System.

Online Services

To locate product-specific information, refer to the following online services:

BBS

To read the Bulletin Board System, set your modem to 8 bits, no parity, 1 stop bit, and dial 508-486-5777 (U.S.).

WWW

The Digital Equipment Corporation Network Products Business Home Page on the World Wide Web is at the following addresses:

North America:	http://www.networks.digital.com
Europe:	http://www.networks.europe.digital.com
Australia:	http://www.digital.com.au/networks

GIGAswitch/FDDI System Firmware 3.1

New Code Images for V3.1

The following files contain **new** firmware images:

File	Firmware Image
fg2-31.rsx	FGL-2 version 3.10
fg4-31.rsx	FGL-4 version 3.10
ag2-31.rsx	AGL-2 version 3.10
ag2p-31.rsx	AGL-2+ version 3.10
scp-31.ftp	SCP version 3.10
sbb-31.ftp	SCP bootblock version 3.10

New MIB

The following files contain the **new** MIB:

File	MIB
mib-31.txt	GIGAswitch MIB version 3.10
e-mib-31.txt	GIGAswitch MIB version 3.10 and DEC ELAN MIB

Version 3.1 Overview and Installation

GIGAswitch/FDDI version 3.1 firmware provides several new features, as well as bug fixes.

New Features

The new features are:

- Reliability Groups
- Learning Domains
- Logical Bridge Domains
- Save and Restore Management Memory
- SNMP Set in OBM

These features are described in the sections that follow.

Installation

Version 3.1 introduces new firmware images for the SCP and all the linecards. Note that a new SCP bootblock image is included with this release. The new boot code is required for the manufacturing and repair process. Although this code does not provide new user functionality, it is recommended that systems be upgraded with this version.

Certain cautions should be observed when upgrading to version 3.0 or 3.1 firmware from pre-3.0 versions. These precautions do **not** apply when upgrading from V3.0 to V3.1.

1) Due to the size of the new SCP firmware image, it cannot be downloaded in a switch running SCP V2.10 or earlier. **SCP V2.20 or later must be running in order to load SCP V3.0 or V3.1 firmware**. SCP V3.1 can be downloaded on a switch running SCP V2.2.

2) The SCP firmware that supports Hunt Groups is not compatible with pre version 3.0 linecard firmware. To avoid compatibility problems, **linecards must be upgraded to V3.0 or later before upgrading the SCP**. If this is not done, all linecards not upgraded will come up in an "FW Rev Mismatch" state when slot configuration is displayed in OBM. Ports in such linecards will not be able to participate in bridging; these linecards will be limited to accepting a new firmware upgrade.

If by mistake the SCP is upgraded before any of the linecards, the switch will be inaccessible by the management station. To restore bridging capability, use the OBM Extended Options Menu (choice 12 from the Main Menu). Choice 3 from the Extended Options Menu allows you to disable/enable Hunt Group support. If you choose to disable Hunt Group support, the switch will reboot and will function normally, except that Hunt Groups cannot be configured. With Hunt Groups disabled, ports on linecards running pre-V3.0 firmware can participate in bridging. After the linecards have been upgraded, the same menu option may be used to restore Hunt Group capability.

NOTE

It is not recommended that the GIGAswitch/FDDI System continue operation with the SCP and linecards running incompatible firmware - even with Hunt Groups disabled.

3) In version 3.0 (and later) of SCP code, the way that MAC addresses are assigned to GIGAswitch/FDDI ports changes. This should have little observable effect - except at the time the V3 image is first loaded. The ARP cache in the network management station (NMS) will have the GIGAswitch/FDDI port's old MAC address assigned to its IP address. If your NMS maintains a database of MAC addresses, you may have to clear the old contents. For more information on how to clear the contents, refer to the documentation for the NMS. If the GIGAswitch/FDDI system is again loaded with pre V3 code, the NMS database should be cleared again.

4) IP address assignments made to the GIGAswitch/FDDI System after the upgrade to V3.0 (or later) will not be visible if the firmware is downgraded to V2.2 or earlier.

New Features in Version 3.1

Reliability Groups

Reliability Groups are variants of Hunt Groups. Whereas Hunt Groups enable multiple physical ports to be grouped into a single logical port for connections between GIGAswitch/FDDI systems, Reliability Groups allow multiple physical ports to be connected to the same ring (or concentrator). The redundant ports act as warm backups; they do not simultaneously carry traffic. This avoids the necessity of a spanning tree reconfiguration to bring redundant ports on line. It is **not** recommended that non-FDDI ports be configured as Reliability Groups.

Create Reliability Groups the same way as Hunt Groups, but set the MIB object **portGroupPortType** to **reliabilityGroup** (2) for Reliability Groups rather than to **huntGroup** (1) (the default) for Hunt Groups.

Domains Overview

Domains are used to keep certain switch activities separated on the basis of port. A domain is defined as a set of logical ports. Domains must be disjoint; a logical port can be in only one domain of a given type. Two types of domains can be defined on GIGAswitch/FDDI Systems: Learning Domains and Logical Bridge Domains. These domains are specified differently, but both serve to keep certain activities restricted to a subset of the logical ports of the GIGAswitch/FDDI System.

Learning Domains

Address Learning

Learning is the process by which an 802.1d bridge such as the GIGAswitch/FDDI System determines where to forward unicast packets. An address is learned to reside on a particular datalink when that address is seen as a Source Address (SA) in a received packet. Once the address has been learned in this manner, 802.1d bridge ports receiving packets for the address may transmit the packets directly to the outgoing datalink on which the address resides. Until then, packets for the address must be flooded on all possible outgoing datalinks.

Restricted Learning

Learning Domains are a means by which the GIGAswitch/FDDI user may limit the scope of 802.1d bridge learning. There are eight learning domains and each logical port is in one and only one Learning Domain. In addition, a logical port may have associated with it up to seven additional Learning Domains, called Target Learning Domains. When an address is learned on a logical port, the information is distributed to all logical ports in the same Learning Domain, and to all logical ports in each of the logical port's Target Learning Domains. The information is not distributed to any other ports.

The effect of this is that after an address has been learned, all ports to which the address information has been distributed begin sending packets for that address to the port where the address was learned. Other ports continue flooding packets for the address to all possible outgoing links. This feature is useful in cases where the same address may be expected to appear on more than one datalink attached to the GIGAswitch/FDDI, such as may occur with a DECnet Phase IV router attached to two GIGAswitch/FDDI ports.

A Learning Domain is specified by listing the logical ports in the Learning Domain. Any logical port that is not included explicitly in a Learning Domain is placed in Learning Domain 1. A port's Target Learning Domains are specified as a list of Learning Domains for each logical port.

Example - Setting Up a Learning Domain

The following example uses MIB objects in the ...**internet.private.enterprises.dec.ema.sysobjid.bridged.gigaswitch.gigaversion1.gigaSets** branch of the GIGAswitch/FDDI MIB to set up a Learning Domain. We define Learning Domain 2 to contain ports 1, 24, and 39.

All domain-related settings are accomplished by utilizing the **Workbuf** MIB objects. These serve as a staging area for settings related to Domains and Hunt Groups. The workbuf stores changes on a temporary basis, allowing all changes to take effect at once.

1) Use **learningDomainMembershipWorkbuf** to assign ports 1, 24, 39 to Learning Domain 2.

```
set  learningDomainMembershipWorkbuf_2 (1,24,39)
```

2) Finalize the operation by replacing the current settings with those stored in various workbuf areas. This also causes management memory to be updated.

```
set portgoupAction  doUpdate
```

If other changes relating to Domains and/or Hunt Groups are being made, this step should not be performed until all **workbuf** areas contain the desired settings.

Example - Setting Up Target Learning Domains

To assign a Target Learning Domain to a port, use the following MIB objects (all but the last are read-only):

...gigaswitch.gigaversion1.gigaSets.portTargetDomainListMembershipTable
.portTargetDomainListEntry
.portTargetDomainListIndex
.portTargetDomainListMembership
.portTargetDomainListMembershipWorkbuf

To assign Learning Domains 3 and 4 as Target Learning Domains of port 1, perform the following SNMP actions:

1) Use **portTargetDomainListMembershipWorkbuf** to assign Learning Domains 3 and 4 as Target Learning Domains of port 1.

```
set portTargetDomainListMembershipWorkbuf_1  (3,4)
```

2) Finalize the operation by replacing the current settings with those stored in the various workbuf areas. This also causes management memory to be updated.

```
set portgroupAction  doUpdate
```

If other changes relating to Domains or Hunt Groups are being made, this step should not be performed until all **workbuf** areas contain the desired settings.

Once this assignment is complete addresses seen on port 1 will be learned by all ports in Learning Domains 3 and 4, as well as by ports in port 1’s own Learning Domain.

Logical Bridge Domains

A Logical Bridge Domain is a set of ports that act together as a logical bridge. This means that both learning and spanning tree activities are restricted to a single Logical Bridge Domain. There can be up to eight (8) Logical Bridge Domains, each operating its own spanning tree, in a GIGAswitch/FDDI System. Initially all ports (1-64) are part of Logical Bridge Domain 1. A Logical Bridge Domain is specified as a list of Learning Domains. The Logical Bridge Domain contains all ports in any of the Learning Domains in the list. Any Learning Domain that is not explicitly included in a Logical Bridge Domain is placed in Logical Bridge Domain 1.

Example - Setting Up a Logical Bridge Domain

All snmp objects for this process can be found under the ....**internet.private.enterprises.dec.ema.sysobjid.bridges.gigaswitch.gigaversion1.gigaSets** branch of the GIGAswitch/FDDI System MIB.

We will define Logical Bridge Domain 6, to consist of Learning Domains 2 and 5, where Learning Domain 2 = ports 1, 24, 39 and Learning Domain 5 = ports 8, 26, 44. Thus Logical Bridge Domain 6 contains ports 1, 8, 24, 26, 39, and 44. In this example the Logical Bridge Domain being defined has 2 constituent Learning Domains. More typically a Logical Bridge Domain would consist of a single Learning Domain.

1) Use **learningDomainMembershipWorkbuf** to assign ports 1, 24, 39 to Learning Domain 2.

```
set learningDomainMembershipWorkbuf_2  (1,24,39)
```

2) Use **learningDomainMembershipWorkbuf** to assign ports 8, 26, 44 to Learning Domain 5.

```
set  learningDomainMembershipWorkbuf_5  (8,26,44)
```

3) Use **LBDomainMembershipWorkbuf** to assign Learning Domains 2 and 5 to Logical Bridge Domain 6.

```
set  LBDomainMembershipWorkbuf_6  (2,5)
```

4) Finalize the operation by replacing the current settings with those stored in various workbuf areas. This also causes management memory to be updated.

```
set  portgoupAction  doUpdate
```

If other changes relating to Domains and/or Hunt Groups are being made, this step should not be performed until all **workbuf** areas contain the desired settings.

NOTE

See the GIGAswitch/FDDI System MIB for detailed information.

MIB Objects within Domains

With the introduction of multiple domains, many MIB objects which were formerly unique (e.g., **dot1dStpPriority**, the spanning tree root priority) become ambiguous. It cannot be determined to which domain (Logical Bridge Domain, in this case) they refer.

A community string-based multiplexing scheme has been added to adequately specify such MIB objects. This mechanism uses the community string and a suffix, separated by a delimiter. A new MIB object, **communityStringDelimiter**, has been created so the user can change the delimiter (in case the default delimiter is contained in the user's community string). The default delimiter is the colon, ":". The set of valid delimiters is the set of printable characters, except for all letters and digits.

A community string suffix gives the SNMP agent additional information to process certain MIB objects. If an objects is not ambiguous, then the suffix is ignored. If an object requires a suffix, but none is supplied, then a default object is assumed. In the case of Learning Domains and Logical Bridge Domains, the default object is the one defined for Domain 1.

There are two types of valid suffixes defined for GIGAswitch/FDDI: LD*x* and LBD*x*, where x is a numeral between 1 and 8 (inclusive). LD3, for example, refers to Learning Domain 3. LBD7 refers to Logical Bridge Domain 7. Both uppercase and lowercase are valid. If the *x* is left out, then 1 is assumed.

If an LDx suffix is used where an LBDy suffix is expected, then the Logical Bridge Domain that contains Learning Domain *x* is used.

If an LBD*x* suffix is used where an LD*y* suffix is expected, then the lowest numbered Learning Domain contained in Logical Bridge Domain *x* is used - if there is one.

Domains and ARP Server

The GIGAswitch/FDDI ARP Server will not observe Domain boundaries in this release. It may respond to an ARP request from one domain with the MAC address of a host in another Domain. In the next release ARP service will respect Domain boundaries, as well as all filters.

Management Memory Save and Restore

GIGAswitch/FDDI System parameters set by the network manager are stored in management memory. This is nonvolatile memory located on the CLOCK card. Each time a management parameter is set (or changed), the setting is recorded in management memory. Whenever the system boots (or a linecard is installed), the relevant parameters are retrieved from management memory, and applied to the running switch.

The management memory save and restore feature allows one to save the current contents of management memory and restore it at a later time to the same or a different switch. This feature is useful for restoring parameter settings in a switch whose CLOCK card has been replaced. It may also be used to assure consistent configurations among a set of switches.

The clearVISN Recovery Manager provides a simple way to perform the save and restore operation without the need to deal directly with SNMP objects.

The following paragraphs explain the details of how the save and restore feature works.

The state of the switch always reflects the contents of management memory, except:

- During the boot process (while the management memory contents is being processed)
- When the contents of management memory are being updated (due to a parameter change)

  or

- While management memory is being restored to a previous state

There are five documented ways that management memory can be altered:

1) **Change a switch parameter** - This occurs whenever a new management parameter is set (or changed) using SNMP or OBM. This causes a new record to be added.

2) **Clear management memory and reboot** - This OBM operation will completely clear the contents of management memory, and cause the system to reboot.

3) **Rewrite management memory** - This OBM operation causes the contents of management memory to be cleared and rewritten, in a compacted form. This is useful when one or more parameters have been changed several times. It eliminates the memory space taken up by all but the last change.

4) **Clear and lock management memory** - This SNMP operation clears management memory and locks out further writes to management memory (except as indicated in (5) below). It is used in conjunction with the restore management memory feature.

5) **Restore management memory** - This SNMP operation causes a previously saved record of management memory contents to be written to management memory after it has been cleared and locked.

The following MIB objects enable one to save and restore the contents of management memory:

...**gigaversion1.gigaBox.clockCard.memoryAction**
...**gigaversion1.gigaBox.clockCard.memoryTable**
...**gigaversion1.gigaBox.clockCard.memoryTable.mgmtMemoryEntry**
...**gigaversion1.gigaBox.clockCard.memoryTable.mgmtMemoryEntry.mgmtMemoryIndex**
...**gigaversion1.gigaBox.clockCard.memoryTable.mgmtMemoryEntry.mgmtMemoryData**

To save the contents of management memory, perform a sequence of **GETNEXT** operations to retrieve all entries of the object **memoryTable**. Each table entry has an index and 80 bytes of data. (The final entry may have less than 80 bytes). These entries must be saved in the order retrieved.

Note that these 80 byte records do not correspond to specific parameters. Parameter records stored in management memory have variable length.

To restore a table of previously saved records:

1) First **SET** the **memoryAction** MIB object to **clear_and_lock** (4). This will clear the contents of management memory, and prevent new data from being written. That means that no parameters may be set or changed until the switch is rebooted.

2) Next perform successive **SET** operations to write the **mgmtMemoryData** for each entry in the table. Use the previously saved records to supply the index and data for these operations.These **SET** operations must be performed in order (as indicated by the index).

3) Reboot the switch to restore the switch to the restored state of management memory.

Note: Writing to nonvolatile memory involves some small degree of risk, since loss of power during such an operation can result in corrupted memory. Loss or corruption of management memory is a rare occurrence, but it is wise to prepare for it.

In the event that management memory is observed to be corrupted, the SCP will not bring up any of the linecards, and will indicate a corrupted management memory on the OBM screen. OBM will offer the opportunity to dump the contents of management memory at that time. To recover, the following steps should be taken:

1) Clear management memory using OBM - the switch will reboot.

2) Assign an IP address to at least one of the ports, so the switch can be managed.

3) Follow the above steps to restore a previously saved parameter configuration - this will include the previously assigned IP addresses. The IP address(es) assigned in step 2 will be gone when the switch is restarted.

or

If management memory has not been previously saved, then all parameters will have to be set again (using OBM or SNMP).

One final note: Although extremely rare, it is possible for management memory to be corrupted in such a way that the SCP finds itself in a loop that causes continual switch reboots. To deal with this unlikely occurrence one must use hardware jumpers located on the CLOCK card. These can be identified by a DIGITAL service representative.

SNMP Set in OBM

Choice 11 in the OBM Main Menu provides access to the values of all supported MIB objects. With this release, OBM can be used to **SET**, as well as **GET**, these values.

New Features in Version 3.0

Descriptions of the features released with V3.0 are included here for completeness. The Hunt Group section has been expanded to include Reliability Groups.

New OBM Menus

The OBM menus have been changed in this firmware release. They have been simplified and made to follow consistent formats.

OBM functions are performed via a series of menus which appear on the OBM terminal. Each menu has a name and a list of choices, preceded by numbers. The user selects the number of the desired choice.

The following conventions are used:

- A choice that ends with the word "menu" results in an additional menu of choices.
- A choice that begins with the word "Show" results in a display, followed by the same menu (or a subset, in the case of the Main Menu).
- A choice that begins with "Show/set" results in a display, followed by a menu of choices related to that display.
- All other choices result in some action, perhaps preceded by input requests or a warning.

  When multiple input values are requested in a single prompt, the values supplied should be separated by spaces.

  When one or more port number is required as one of the inputs, the FPPN numbering scheme should be used in one of the following formats:

  single port: (10.4)

  list of ports: (10.1,10.2,12.2,14.1)

  range of ports: (10.1-12.4)

  "all" - to indicate all ports

- The final choice for all menus (except for the Main Menu) returns to a previous menu-often to the Main Menu. The final choice in the Main Menu ends the OBM session.
- Pressing <CR> causes a previous menu or the main menu to reappear, except as indicated below.
- A display that cannot be shown on a single screen will have "(<cr>=More)" following the prompt. Press <CR> to continue the display.
- <Ctrl/D> will abort the OBM session at any time.

The first menu presented by the OBM is called the Main Menu. The Main Menu consists of the System, Port, Bridge, MIBs, Extensions, and OBM sections.

```
         Main Menu

System      1. Show box configuration

            2. Show/set slot configuration

            3. Show/set system time

            4. Clear management memory

            5. Reboot menu

Port        6. Show LAN address assignments

            7. Show/set privileged and bootserver ports

            8. Show/set delayed ports

            9. IP menu

Bridge     10. Bridge menu

MIBs       11. MIB viewer menu

Extensions 12. Extended options menu

OBM        13. OBM menu

           14. End OBM Session

Choice:
```

Choices made from the Main menu provide similar functionality to corresponding choices in the OBM menus in previous versions of GIGAswitch/FDDI firmware. For more information on the new OBM features refer to the *GIGAswitch/FDDI System Out-Of-Band Management (OBM) Guide (*EK-GOBMG-MG).

24K Translation Table Size

Version 3.0 allows one to choose a translation table (TT) size of up to almost 24,000 MAC addresses. There are now 4 choices of maximum table size: 3,737, 7,737, 15,737 and 23,993. The desired size is selected using the OBM Bridge Menu, which is choice 10 on the Main Menu.

The choice of translation table size determines the maximum number of MAC addresses the SCP will send to any port. If the firmware running on a linecard does not support the chosen maximum size, the SCP will not allow its ports to be brought on line. It will indicate an "FW Rev Mismatch" status in the OBM slot display. This condition can be resolved by either downloading appropriate firmware to the linecard in question or by resetting the Translation Table size to a lower number.

The AGL-2 does not support 24K table size. It will always show an FW Rev Mismatch in the OBM slot display when booted with 24K size in effect. Hence, the TT size must be set to a lower number in order to have an active AGL-2 port. The AGL-2+ linecard does support 24K table size. The default TT size is 7,737.

Demand Learning

Demand Learning is a feature that reduces the learning activity on certain ports. Ordinarily when a new source address (SA) is seen on a port, the SCP informs every GIGAswitch/FDDI port of the association between that address and the port on which it was seen. Thus the entry takes up space in every port's forwarding table.

When demand learning is enabled, the SCP initially only notifies the port on which the address was seen. This is critical, since that port is responsible for aging the address, and it needs to know that packets destined for that address should not be forwarded through the crossbar.

Other ports are not notified until they "need" to be. If a packet arrives at a port, destined for that particular address, this port will send the packet to the SCP for flooding (since, for this port, it is an unknown address). The SCP will then realize that this port has a "need" to know that address. So, in addition to flooding the packet, it will convey the information about the address to this port. The next time this port sees the address, it will know which port to send it to.

This feature has the advantage of reducing the number of entries in each port's forwarding table, which conserves space and reduces the overhead of maintaining the table. It has the disadvantage of causing an extra flooding event the first time each port sends to each address. If every port will eventually send packets to most addresses, then demand learning saves little, and consumes extra overhead. If most ports send only to a limited number of addresses, then the overhead of this feature may well be worth expending for the improved capacity/performance effects.

It will require some analysis, or even experimentation to determine whether this feature is appropriate at a given site.

The demand learning feature is set up using the following MIB object:

...**gigaversion1.gigaBridge.gigaStp.gigaStpDemandLearningEnable**

Set this object to **True** (1) to enable demand learning, and to **False** (2) to disable demand learning. It is set to **False** (2) by default.

AGL-2+ Module

The AGL-2+ module is supported in this release. Operationally, the AGL-2+ has the same features as its predecessor, the AGL-2. It differs only in the following ways:

- The AGL-2+ will support translation table size up to 23,993. The AGL-2 only supports up to 15,737 addresses.
- The AGL-2+ uses new modPHY daughter cards instead of the daughter cards used on the AGL-2. These modPHYs can be installed without removing the module from the chassis. **However they should not be installed while the module is powered on**. Instructions for installation are included with the modPHY.
- Interconnections between AGL-2 and AGL-2+ are supported.

For more information on the new AGL-2+ features, refer to the *GIGAswitch/FDDI System AGL Reference Guide* (EK-GAGL2-MG).

Hunt Groups

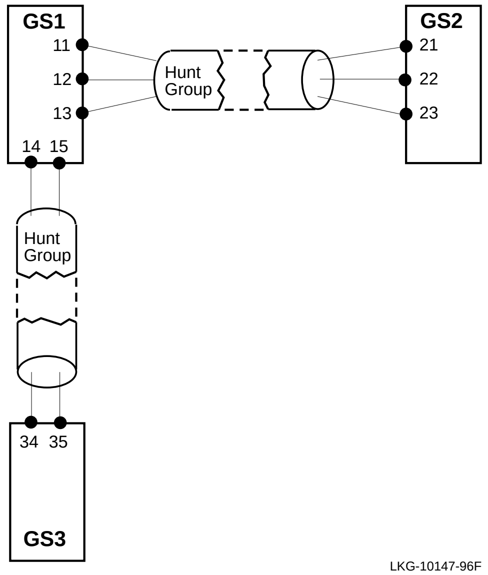
The Hunt Group feature allows a pair of GIGAswitch/FDDI Systems to communicate over two or more active links. This is accomplished by configuring two or more physical ports as a Hunt Group. When this is done the switch treats that group of ports as a single bridge port. Spanning tree, learning, aging and filtering all see the Hunt Group as a single port. A source address seen on one of these ports is associated with the Hunt Group port, not the physical port; spanning tree places the Hunt Group bridge port, not the individual ports, into forwarding or blocking state; addresses are aged on the entire Hunt Group, not on an individual port; and filters are applied to the Hunt Group, not the individual ports.

In the following figure a 3-member Hunt Group joins GS1 to GS2, and a 2-member Hunt Group joins GS1 to GS3.

Configuring several physical ports as a Hunt Group offers two advantages:

- It allows a higher rate of traffic flow between two GIGAswitch/FDDI Systems.
- It provides quick failover in the event of a link or port failure.

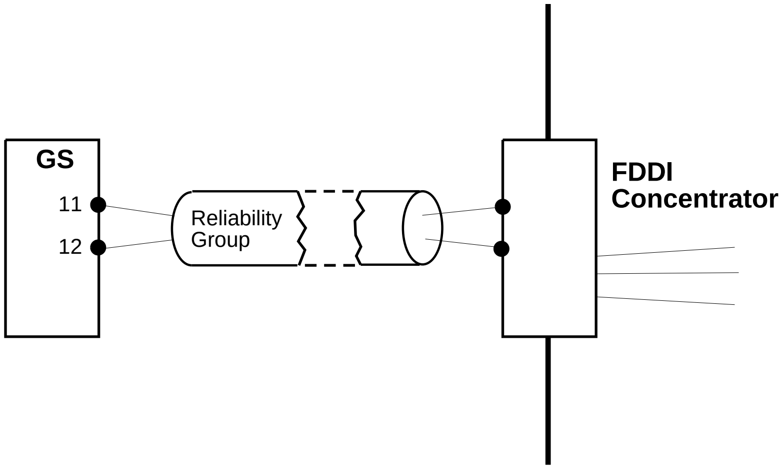
The following section contains more detailed information about Hunt Groups.



LKG-10147-96F

Reliability Groups

The Reliability Group feature allows a single GIGAswitch/FDDI System to connect to an FDDI ring or concentrator over two or more ports, without placing the redundant ports in spanning tree backup state. Unlike Hunt Groups, only one of the Reliability Group members carries data at a time. This feature is purely a fast failover feature. Whereas spanning tree failover is typically 30 to 50 seconds, failover to a standby Reliability Group member will be accomplished in 2 to 4 seconds.



LKG-10316-96F

Hunt Groups

Hunt Group Member Ports

A physical port configured in a Hunt Group must:

- Be connected in a point-to-point link
- Run in full duplex mode
- Be connected to a port (on another GIGAswitch/FDDI System) which is also configured as part of a Hunt Group

Whenever a physical port is configured in a Hunt Group, the SCP regularly sends proprietary protocol messages through that port. Once the two switches at opposite ends of the link agree that they are at opposite ends of a Hunt Group, the Hunt Group is established. As additional ports are identified as belonging to the same Hunt Group, the Hunt Group is reconfigured. Hunt Group Member Numbers are assigned as members join the Hunt Group. Hunt Group Member Number 1 is assigned to the member with the lowest physical port number, 2 to the next lowest, etc.

Hunt Group Port Numbers

Hunt Groups appear to the GIGAswitch/FDDI System as new ports, with different port numbers than ordinary ports. There are two numbering schemes used to refer to ordinary ports, SPN (sequential port number) and FPPN (front panel port number). Both of these schemes have been extended to refer to Hunt Groups as well. The SPN of an ordinary port can range from 1 to 36. The SPNs of Hunt Groups range from 37 to 64. The FPPN of an ordinary port can range from 1.1 to 14.2. The FPPNs of Hunt Groups range from 99.37 to 99.64. In Hunt Groups the FPPN does not have any physical meaning, as it does for ordinary switch ports.

With the addition of Hunt Group ports, it becomes necessary to distinguish the different uses for port numbers. Port numbers refer to both physical and logical ports (or bridge ports). Prior to the existence of Hunt Groups, the physical port number and the logical port number were identical for a given port. For Hunt Groups, that will no longer be the case. Two physical ports may have different media or other characteristics, even though they belong to the same Hunt Group. The logical port number is used for all bridge operations: spanning tree, learning, aging, filtering. Note that SPNs or FPPNs can both be used to refer to either logical or physical ports. The following discussion uses SPNs.

Logical Ports

Every bridge entity in the GIGAswitch/FDDI System (e.g., learning, aging, filtering, and spanning tree process) deals with logical ports. The GIGAswitch/FDDI System has 64 logical ports, with SPN 1 to 64. Logical ports are also known as bridge ports. The ports that are actually placed in the box are called physical ports. The switch can access up to 36 physical ports, with SPNs from 1 to 36.

Logical ports can operate only after some physical ports are assigned to them. Since there are more logical ports than physical ports, some logical ports must have no physical port assigned to them. These logical ports are called "empty" ports. Empty ports do not participate in bridge functions.

In the default system configuration, each logical port between 1 and 36 is associated with one and only one physical port, and the mapping from physical ports to logical ports is the identity mapping. Namely, physical port *n* is assigned to logical port *n*, where 1 <= *n* <= 36. A logical port to which only one physical port is assigned is called a singleton bridge port. Logical ports 37 to 64 are empty ports in the default configuration. Hunt Groups are created by assigning 1 or more physical ports to logical ports in the range 37 to 64. Any physical port which is not so assigned retains (or reverts to) its default assignment.

Hunt Group Example

To create a Hunt Group consisting of physical ports 1, 3, and 5, choose a logical port number by which to refer to this Hunt Group. The logical port number must be in the range 37 to 64. Suppose logical port 45 is chosen. Assign physical ports 1, 3, and 5 so they belong to logical port 45 (45:{1,3,5}). When this is done the logical ports 1, 3, and 5 become empty logical ports.

The three physical ports that are now logically assigned to port 45 must next be connected to three ports on a second GIGAswitch/FDDI System. The ports they're connected to must be assigned to a Hunt Group as well. There is no requirement that the logical port numbers be the same on both switches.

Learning

A packet entering the switch through a physical port configured in a Hunt Group has its source address (SA) learned on the Hunt Group logical port. In the above example a packet entering the switch through physical port 1 would have its SA learned on port 45. The forwarding tables in the SCP and on all linecards will list port 45 as the home of that MAC address.

Aging

With several physical links connecting two switches, the aging process becomes more complicated. This complexity has been addressed by having the two GIGAswitch/FDDI Systems communicate aging information explicitly. A GIGAswitch/FDDI System will not by itself age out addresses seen on a Hunt Group port. It will age them out only after the switch at the other end of the Hunt Group links has aged them out.

This will result in an additional delay in aging, but it will not affect network operation significantly.

Filtering

Filters (both dynamic and management-set) apply to logical ports. It is important to note that previously defined filter matrices (including the default filter matrix) may prevent traffic from traversing a Hunt Group port, since Hunt Group ports all have SPN greater than 36. So be sure to examine, and, if necessary, modify existing filter matrices to be certain they account for new logical ports introduced by Hunt Groups.

Single-Path and Multi-Path Packets

When a multi-member Hunt Group exists, there are multiple paths between certain points on the network. Since a packet may traverse any of the Hunt Group's member links, it has more than one way of going from a station on one side of the Hunt Group to a station on the other side. Hence, there is the possibility that a stream of packets from a given source to a given destination could arrive out of order.

For some protocols and applications, out-of-order packet arrival is acceptable. For others it is unacceptable. The GIGAswitch/FDDI System can differentiate packets as single-path or multi-path, based on incoming physical port and protocol type. In a later section we explain how a network manager can specify which packets are identified as single-path and which as multi-path. As the name suggests, single-path packets from a given source to a given destination will be guaranteed to traverse a single path. Hence they will arrive in order. Multi-path packets may traverse different paths between a source and a destination. Hence, they may arrive out of order. In later sections we describe how load balancing is performed for single-path and multi-path packets.

Out-of-Order Packets

While most transport protocols are designed to handle out-of-order packet delivery, there are some implementations of such protocols that have been observed to fail when significant numbers of packets arrive out of order. Furthermore, there are some protocols, Local Area Transport (LAT) for instance, that are not designed to handle out-of-order packet delivery at all. The network manager can designate a protocol to be "single-path" in order to adjust to such circumstances. By default, the GIGAswitch/FDDI System assumes all protocols are single path, except for: DECnet, IP, IPX, NISCA, all of which are pre-defined as multi-path.

TCP and Out-of-Order Packets

Some implementations of TCP may be tolerant of, but sensitive to out-of-order packet delivery. One such example is DIGITAL UNIX TCP/IP. This implementation employs a "fast retransmit" algorithm wherein the receiver of an out-of-order packet immediately sends a (duplicate) ACK for the last in-sequence packet received. When the number of duplicate ACKs exceeds a certain threshold, the sender considers a packet to have been dropped, and retransmits, even if the rexmt timer has not yet expired.

The threshold is kept in a kernel variable called tcprexmtthresh. The default value for this variable is 3, which means that after receiving three duplicate ACKs caused by three out-of-order packets, the sender retransmits. If many such retransmissions occur, application performance or available bandwidth could be adversely affected. It is recommended that the value of this variable be increased for DIGITAL UNIX systems if a large number of retransmissions are observed. Increasing it to 100 should eliminate such unnecessary retransmissions caused by out-of-order packet arrival.

The following commands will accomplish this:

Commands	Comments
% dbx -k /vmunix	Invoke the dbx debugger.
(dbx) p tcprexmtthresh 3	Print the current value Current value is 3.
(dbx) assign tcprexmtthresh=100 100	Assign new value of 100.
(dbx) patch tcprexmtthresh=100 100	Patch image on disk, so value 100 will be used the next time the system reboots.

Dynamic Load Balancing

Referring back to our example of a three member Hunt Group, recall that logical port 45 has three physical ports (1, 3, and 5) assigned to it. When a multi-path packet arrives at the switch, destined for an address seen on port 45, it will be sent to the first of the ports 1, 3, or 5 that is available to take traffic from the crossbar. This results in the most efficient use of the crossbar bandwidth. But, as noted above, it can result in packets being delivered out of order.

Note that actual load balancing need occur only when the crossbar connection to one of the Hunt Group members is busy. In the absence of crossbar congestion, all traffic may flow through a single physical port.

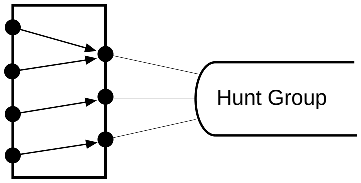
But lack of congestion at the crossbar does not necessarily imply lack of congestion on the outbound links. Dynamic load balancing cannot ensure that the external links are load balanced.

As long as all ports in the Hunt Group can deliver frames to the outbound link at the crossbar speed (100 Mb/s), there should not be a problem. But a DS3 port, for example, cannot keep up with the crossbar bandwidth. Hence when a DS3 port is a Hunt Group member, dynamic load-balancing may not work efficiently. It is not recommended that DS3 links be used in Hunt Groups as a means of increasing bandwidth for multi-path packets. But static load-balancing (described below) works properly for DS3 links.

And DS3 links can be effectively configured as part of a Hunt Group to provide a redundant data path with enhanced failover capabilities. The reason we require Hunt Group member ports to operate in Full Duplex mode is to assure they can provide 100 Mb/s output bandwidth.

Static Load-Balancing

Static load balancing is applied to single-path packets, and works as follows: Under stable conditions, all single-path packets received on any one GIGAswitch/FDDI inbound physical port are transmitted on the same outbound physical port of any given Hunt Group. However, single-path packets received on different inbound physical ports may be configured to use different outbound physical ports in a Hunt Group. In this way the load may be spread over the Hunt Group members. The network manager controls the mapping of inbound physical ports to outbound Hunt Group members, as indicated in the following figure.



LKG-10143-96F

The advantage of static load balancing is that packets are delivered between any source address and any destination address in the order in which they are sent, except in the rare event of network reconfigurations. This contrasts with dynamic load balancing, in which packets are routinely delivered out of order.

The disadvantages of static load balancing are these: First, the load balancing is not as efficient as dynamic load balancing, since the assignment of a packet to a Hunt Group transmit link does not take into account dynamic port loading. Second, to make the best use of the feature, the network manager must know switch traffic patterns and must configure each Hunt Group carefully to spread the load evenly over its members.

Finally, when Hunt Group member links come up or down, there is a possibility that a few packets may be delivered out of order. In-order packet delivery may be absolutely guaranteed, by use of **fixed** traffic category described below, which eliminates load balancing entirely.

Note: While static load balancing will assume in order delivery of unicast packets, the GIGAswitch/FDDI System cannot assume that a stream consisting of both unicast and multicast packets will arrive in order - whether or not a Hunt Group is in the path.

Traffic Groups

When a packet arrives at a GIGAswitch/FDDI port, the receiving linecard classifies it as single-path or multi-path, according to its service class and port number. A later section describes how to configure service class and ports to properly classify packets.

The network manager may assign a physical port to one of 16 traffic groups (numbered 1-16). By default, all ports (1-36) are assigned to traffic group 1. A single-path packet entering a port is assigned that port's traffic group number. It is said to belong to that traffic group.

All single-path packets belonging to the same traffic group and destined to the same Hunt Group will exit the switch through the same physical port. For each Hunt Group, the network manager designates a traffic category and a Hunt Group member number per traffic group. (Recall that the physical links of a Hunt Group are numbered from 1 to the current Hunt Group size by "Hunt Group member number").

The designated traffic category and Hunt Group member number determine the physical out bound link as follows:

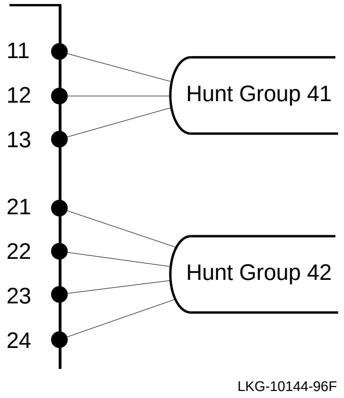
- If the traffic category is **reconfig**, the packet is transmitted on the link that corresponds to the designated Hunt Group member number.
- If the traffic category is **fixed**, the packet is always transmitted on one selected Hunt Group link. A different link is selected only if the currently selected link leaves the Hunt Group (for example, if it fails). The Hunt Group member number is ignored in this case.

  If a designated Hunt Group member number, *n*, exceeds the number of active Hunt Group members, *k*, the number (*n* mod *k*)[1] is used instead. Whenever a physical port joins or leaves a Hunt Group, the numbers assigned to members often change, and the mapping of traffic groups to members will change accordingly. But recall, that this applies only to **reconfig** traffic category.

Illustration of Static Load-Balancing

Consider a GIGAswitch/FDDI system configured by the network manager as follows:

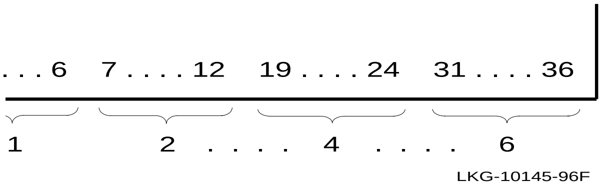
- Ports 11, 12, and 13 comprise a Hunt Group with bridge port number 41 (41:{11,12,13}).
- Ports 21, 22, 23, and 24 comprise a Hunt Group with bridge port number 42 (42:{21,22,23, 24}).



[1] *n* mod *k* = remainder after diving *n* by *k*

- The memberships of traffic groups 1 through 6 are configured according to the following table. Traffic groups 7-16 have no ports.

Traffic Group	**Ports**
1	1, 2, 3, 4, 5, 6
2	7, 8, 9, 10, 11, 12
3	13, 14, 15, 16, 17, 18
4	19, 20, 21, 22, 23, 24
5	25, 26, 27, 28, 29, 30
6	31, 32, 33, 34, 35, 36



- Traffic groups are mapped by the network manager to member numbers in Hunt Group logical ports 41 and 42 as shown in the following table:

Traffic Group	**Hunt Group 41 Member Number**	**Hunt Group 42 Member Number**
1	1	1
2	2	2
3	3	2
4	1	3
5	2	3
6	4	4

Packets from traffic group 1 leave through Hunt Group member 1 of both Hunt Groups, 41 and 42.

Packets from traffic group 2 leave both Hunt Groups through member 2.

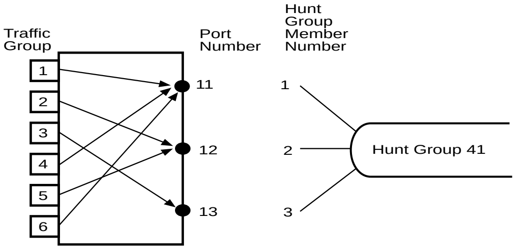
Packets from traffic group 3 leave Hunt Group 41 through member 3, and leave Hunt Group 42 through member 2.

Packets from traffic group 4 leave 41 through member 1, and leave 42 through member 3.

Packets from traffic group 5 leave 41 through member 2, and leave 42 through member 3.

Packets from traffic group 6 leave 41 and 42 through member 4. But Hunt Group 41 has only 3 members. So this traffic group leaves Hunt Group 41 through member 1, which is (4 mod 3).

Single-path traffic leaving on bridge port 41:



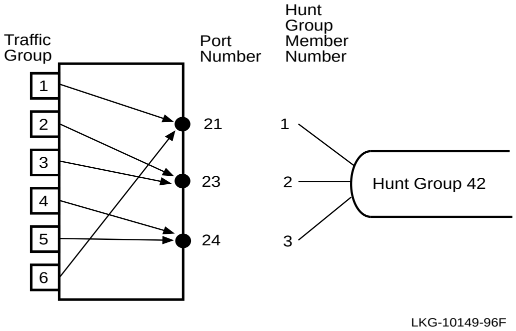
LKG-10148-96F

Traffic Group	Inbound Physical Ports	Hunt Group Member Number	Physical Port Number
1, 4, 6	1-6, 19-24, 31-36	1	11
2, 5	7-12, 25-30	2	12
3	13-18	3	13

Single-path traffic leaving on bridge port 42:

Traffic Group	Inbound Physical Ports	Hunt Group Member Number	Physical Port Number
1	1-6	1	21
2, 3	7-12, 13-18	2	22
4, 5	19-24, 25-30	3	23
6	31-36	4	24

If physical port 22 were to fail, there would be only three Hunt Group members in Hunt Group 42 (ports 21, 23, and 24). Port 23 would become member number 2, and port 24 would become member number 3. All the traffic that had been going to member number 4 would now be handled by member number 1 (4 mod 3), which is port 21. The new traffic pattern would be as follows:



Single-path traffic leaving on bridge port 42 after failure of port 2:

Traffic Group	Inbound Physical Ports	Hunt Group Member Number	Physical Port Number
1, 6	1-6, 31-36	1	21
2, 3	7-12, 13-18	2	23
4,	19-24, 25-30	3	24

Hunt Group MIB Objects

An updated revision of the GIGAswitch/FDDI MIB is required to manage Hunt Groups. The following objects, part of the gigaSets branch of the MIB, are used to manage Hunt Groups.

portGroupMembershipTable
- **portGroupBridgePort**
- **portGroupMembership**
- **portGroupMembershipWorkBuf**
- **portGroupPortType**
- **portGroupPortTypeWorkBuf**
- **portGroupPortOperStatus**

portGroupMembershipFppnTable
- **portGroupFppnBridgePort**
- **portGroupFppnMembership**
- **portGroupFppnMembershipWorkBuf**
- **portGroupFppnPortType**
- **portGroupFppnPortTypeWorkBuf**
- **portGroupFppnPortOperStatus**

portGroupStatusTable
- **portGroupStatusBridgePort**

portGroupStatusPortNumber
portGroupStatusPortType
portGroupStatusOperStatus

portGroupAction

portGroupActionStatus

The **portGroupMembershipTable** is used to configure Hunt Groups. It is indexed by bridge port numbers, in the range 37-64. The **portGroupMembership** field contains a specification of the physical ports assigned to the indexed Hunt Group. The **portGroupPortType** field specifies whether the port group is a Hunt Group or a Reliability Group. It has value **portGroup** (1) or **reliabilityGroup** (2).

The **portGroupMembership** objects are not directly settable. They are read-only. Instead the **portGroupMembershipWorkBuf** objects are set using SNMP. When all the **WorkBuf** objects have the desired values, set the **portGroupAction** to **Update** (2), to cause the changes in all **WorkBuf** fields to take effect in a single operation.

The following status indicators can be used to find out various states of each port:

- **portGroupPortOperStatus** indicates whether a Hunt Group port is initialized and works as a logical port.

- **portGroupStatusTable** provides various information about physical ports such as Hunt Group memberships and operational states.

- **portGroupActionStatus** indicates the status of the last **portGroupAction** operation.

Hunt Group Configuration Example

This section demonstrates how to use these MIB objects to create the Hunt Group of the earlier example: Hunt Group logical port number 45 has 3 members, physical ports 1, 3 and 5. The Hunt Group is set up in steps 1 through 4 below. It is torn down in step 5.

Note: To perform the same operations using FPPN notation for ports, use the **portGroupMembershipFppnTable**.

1) Set **portGroupMembershipWorkBuf** to a string describing the members.

```
set portGroupMembershipWorkBuf_45     (1,3,5)
```

The instance variable of the object is logical port number 45. It is the same as the Hunt Group port number. The string is of the form (<list>), where <list> is a port list specification. A port list specification is a sequence of port numbers (spn) or port ranges, separated by commas. A port number is a number between 1 and 36 inclusive. A range is of the form *a-b*, where *a* and *b* are port numbers, and *a* < *b*. This range specifies all numbers between and including *a* and *b*. Thus (1-3,5,7-36) specifies every physical port except ports 4 and 6. Specifying a null list, (), removes all ports from the Hunt Group.

2) Get **portGroupMembershipWorkBuf** object and see that the value is correctly set. If not, repeat step (1).

```
get portGroupMembershipWorkBuf_45
```

(Object **portGroupMembership**_45 has the value that is currently effective. This will not show any change until the next step).

3) Set **portGroupAction** to **doUpdate** (2) to make the set effective. After the successful completion of this SNMP set operation, the system will execute the following actions:

— Copy **portGroupMembershipWorkBuf_**45 to **portGroupMembership_**45

— Permanently record the changes in the system management memory.

— Initialize Hunt Group port 45 (i.e., update physical ports 1, 3 and 5 as members of logical port 45).

As a side effect, logical ports 1, 3, and 5 become empty ports.

Note: Do not set **portGroupAction** to **doUpdate** until all Hunt Groups have been specified as desired.

The above copy operation will be performed on all indices for which **portGroupMembershipWorkBuf** is different than **portGroupMembership**.

4) Get **portGroupMembership** and see if it is identical with its corresponding work buffer.

get portGroupMembership_45

Hunt Group 45 has now been created.

5) To tear down Hunt Group port 45, the following steps must be taken:

set **portGroupMembershipWorkBuf_**45 ()

set **portGroupAction doUpdate**

After the successful completion of these two sets, Hunt Group 45 has been torn down. Logical port 45 becomes an empty port and logical ports 1, 3, and 5 become singleton bridge ports.

MIB Objects for Static Load-Balancing

The following objects, in the **gigaSets** branch of the MIB, are used to configure static load balancing on Hunt Groups:

trafficGroupMembershipTable
  trafficGroupNumber
  trafficGroupMembership

trafficGroupAttributeTable
  trafficGroupNum
  trafficGroupHgNumber
  trafficGroupHgMember
  trafficGroupCategory

The **trafficGroupMembershipTable** is a table that specifies the physical ports that are assigned to each of the 16 traffic groups. It is indexed by the **trafficGroupNumber**, a number in the range 1-16. Each entry has an object, **trafficGroupMembership**, whose value is a string that specifies the physical ports assigned to that traffic group. The string is of the form (<list>), where <list> is a sequence of port numbers or ranges of port numbers, separated by commas. A port number is a number between 1 and 36 inclusive. A range is of the form *a*-*b*, where a and b are port numbers, and *a* < *b*. This range specifies all numbers between and including *a* and *b*. Thus (1-3, 5, 7-36) specifies every physical port except ports 4 and 6. When a traffic group is specified, any physical port included in the specification is removed from its previous traffic group. Any physical port that is not explicitly defined to be in a traffic group defaults (or reverts) to traffic group 1.

The **trafficGroupAttributeTable** is a table that specifies how traffic groups are allocated to Hunt Group members. It is indexed by **trafficGroupNum** (a number in the range 1-16), and **huntGroupHgNumber**, a number in the range 37-64. Each entry has two associated objects:

- The **trafficGroupCategory**, has one of the following values:
  - — **fixed** - causes all traffic to exit the switch through a single Hunt Group member, which changes only if the current one leaves the Hunt Group. This will guarantee in-order packet delivery in all cases, but provides no load balancing.
  - — **reconfig** - assigns traffic to Hunt Group members, as specified by the trafficGroupHgMember. Traffic is spread among Hunt Group members, and in order delivery is guaranteed, except during a Hunt Group reconfiguration.
- **trafficGroupHgMember**, which specifies the Hunt Group member number of the port which will transmit packets from this traffic group. This is a number in the range 1-16.

Static Load-Balancing Example

Setting up the load balancing for Hunt Group 41, with member ports 11, 12, 13 are described in the examples starting on page 19.

First we define traffic groups 1, 2, 3, 4, 5, and 6:

set **trafficGroupMembership**_1 (1-6)
  trafficGroupMembership_2 (7-12)
  trafficGroupMembership_3 (13-18
  trafficGroupMembership_4 (19-24)
  trafficGroupMembership_5 (25-30)
  trafficGroupMembership_6 (31-36)

Next, in the **trafficGroupAttributeTable**, for Hunt Group 41, and each traffic group, 1, 2, 3, 4, 5, and 6, set the value of **trafficGroupCategory** to be reconfig.

```
set trafficGroupCategory_1.41   reconfig
    trafficGroupCategory_2.41   reconfig
    trafficGroupCategory_3.41   reconfig
    trafficGroupCategory_4.41   reconfig
    trafficGroupCategory_5.41   reconfig
    trafficGroupCategory_6.41    reconfig
```

Finally assign the values of **trafficGroupHgMember** according to the table on page 20.

```
set trafficGroupHgMember_1.41   1
    trafficGroupHgMember_2.41   2
    trafficGroupHgMember_3.41   3
    trafficGroupHgMember_4.41   1
    trafficGroupHgMember_5.41   2
    trafficGroupHgMember_6.41    4
```

MIB Objects to Specify Single-Path Packets

The following MIB objects are used to specify which packets are to be classified as single-path packets. Packets are classified as single-path or multi-path according to the protocol type. So a packet's SAP or SNAP value is the key to its classification. Recall single-path packets are the packets that are governed by static load balancing across Hunt Group member ports. All of these objects are part of the **.gigaswitch.gigaversion1gigaBridge.ServiceClassAssignments** branch of the MIB.

```
ebrNportSnapSvcTable
    ebrNportSnapSvcSnapValue
        ebrNportSnapSvc
        ebrNportSnapSinglePath
       ebrNportSnapSvcStatus

ebrNportSapSvcTable
    ebrNportSnaSvcSnaValue
        ebrNportSapSvc
        ebrNportSapSinglePath
       ebrNportSapSvcStatus
```

These tables are indexed by SNAP and SAP values, respectively. A particular protocol may be established as a single-path protocol by setting the value of its **ebrNportSapSvc** or **ebrNportSnapSvc** to be 0. Setting the value to 1 makes this a multi-path protocol. The GIGAswitch/FDDI System comes with certain protocols preconfigured as multi-path. They are: IP, IPX, NISCA, ARP, DECnet Phase IV, ISO CNLS. All other protocols are, by default, single-path. Use the above objects to change any of these default settings.

The **ebrNportSnapSinglePath** and **ebrNportSapSinglePath** fields of the table may be set with a port list. On these ports the corresponding protocol will be treated as single-path, even though it may be defined (using the ..SvcTable) as a multi-path protocol. In short, this is a way to override a multi-path designation on a select list of ports.

Single-Path Protocol Example

To change the service class of a protocol to be different from the default values one must create an entry in the **erbNportSnapSrvTable** MIB object. This table is indexed by SNAP value. For example, the SNAP value for IP is 00-00-0008-00. So the entry corresponding to IP will be indexed by 0.0.0.8.0. By default, IP is set as a multi-path protocol. If one wishes to set IP to be a single-path protocol, set the MIB object:

...**gigaBridge**
.**ServiceClassAssignment**s
.**ebrNportSnapSvcTable**
.**ebrNportSnapSvcEntry**
.**ebrNportSnapSvc.0.0.0.8.0**

to have value 0 (for single-path). Then set the status of this entry to be "permanent" by setting the MIB object:

...**gigaBridge**
.**ServiceClassAssignments**
.**ebrNportSnapSvcTable**
.**ebrNportSnapSvcEntry**
.**ebrNportSnapSvcStatus.0.0.0.8.0**

to be 2 (permanent).

Setting the status of an entry to be "invalid" (value = 1) will invalidate that entry. The service class of the corresponding protocol will revert to the default setting.

To set the service class of a protocol to be "multi-path," use the value 1, in place of 0, above.

Problems Resolved in Version 3.0

The following problems, seen in previous releases, were fixed in V3.0, as described below.

AGL-2 Fast Firmware Download

The time required for firmware download of an AGL-2 module has been reduced from 15-20 minutes to just over 2 minutes. The new download process requires that the latest versions of firmware be running on both SCP and on AGL-2. Hence, the time savings will not be observed until the succeeding firmware version is loaded. The released firmware for AGL- 2+ includes the faster download process, but field test and pre-released firmware versions do not.

Full Duplex Operation

In previous versions, a port would occasionally drop out of full duplex mode. Such drops will no longer occur.

Flooding Performance

A shared buffer scheme and other optimizations have been added to improve flooding performance.

FGL-4 in FGL-2 Slot

Fixed a problem that caused an FGL-4 to crash when placed in an FGL-2 slot.

ifOperStatus

This SNMP object now reports the proper value.

Backup SCP

Fixed problem that caused SCP to go through initialization twice when it became a backup SCP.

Short Entry Age

Fixed bug whereby SCP would sometimes not switch to the short aging timer when required (e.g., if a link goes down).

Flooding Counters

Changed flooding counters type from INTEGER to COUNTER, to prevent them from appearing to be negative numbers.

M-port Failover

Speeded up the wrongSA scan to reduce the time required for M-port failover.

Linecard Error Logs

Linecard error logs now only display valid entries, and the entries include the firmware version.

PMD LEDs

When an FDDI port is disabled via the **snmpFddiPortAction** MIB object, the left PMD led will now blink green to indicate the port has been disabled by management.

IP Packet Reassembly

IP packets directed to the GIGAswitch/FDDI System are now reassembled correctly.

Downline Load Indicator

While the FGL-4 FLASH is being written during a firmware download, the module led will blink green.

FGL Port Indexing

Several problems related to the reporting of port information in SNMP have been corrected.

Forward Delay

The value of ForwardDelay was reported by SNMP as twice the actual value. This is now reported correctly.

Spanning Tree Parameters Change

If the spanning tree root's parameters were changed during a preforwarding delay, the change did not take effect until after the preforwarding delay. Now it properly takes effect immediately.

Improper TCN

Topology change notifications would occasionally be sent at inappropriate times. This no longer occurs.

Cut-Through

Cut-through is now properly enabled and reported on all ports.

Forwarding State of M-ports

Once an M-port enters the forwarding state, it will not leave it - except during a firmware download or when disabled by the SCP.

Problem Resolved in Version 3.1

The following problems have been fixed in version 3.1 of the firmware.

Filter Induced SCP Panic

The GIGAswitch/FDDI SCP would crash when filters were set on some specially treated addresses.

Ring Purger Causes Performance Problems - Ring Errors

When ring purger was enabled, the FGL would occasionally not initialize properly. The observed results were poor performance and bridge strip errors.

Short Aging Time Not Used After Topology Changes

After topology changes, the linecards would continue to use the normal aging timer.

ebrNportPortNum Settings Can Cause SCP Crash

If **ebrNportPortNum** or **ebrNportFppnPortNum** SNMP objects were set prior to the BL3.0 release, upgrading to BL3.0 could cause repetitive SCP crashes or improper behavior. Upgrading to V3.1 will avoid this behavior. Such objects set prior to V3.0 will be ignored. Ignored records can be displayed from the OBM Extended Options Menu (Management Memory deletions).

ARP Server Flooding

The GIGAswitch/FDDI system would erroneously flood ARP requests when the ARP server was enabled.

AGL Counters

Traffic counters for AGL ports would report incorrect data.

Elasticity Buffer Errors Cause PHY Reset

FGL-2 or FGL-4 would reset the PHY when elasticity buffer errors were encountered. This would occasionally prevent the FDDI ring from coming up when connected to devices of certain other vendors, because of some minor timing problems which cause elasticity buffer errors.

FGL Reboots and 114 and/or 118 (Missing Start Of Packet) Error Log Entries

When subjected to large numbers of bad frames, the FGL would occasionally reboot and/or log errors of type 114 or 118.

Aging Scan Improvements

The time required by the linecards to note and report aging events has been improved. This enables aging times of less than 1 minute to be implemented more predictably.

ifInUcastPkts Is Sometimes Invalid for Second Port on a Linecard

MIB counters on a linecard's second port were sometimes invalid.

Configuration BPDUs

The SCP would erroneously send configuration BPDUs when spanning tree parameters were changed.

Oversized SMT Frames

Oversized SMT frames would occasionally cause FGLs to reboot.

OBM Hang

Bugs in some OBM menus could cause the system to hang if an improper input was typed.

Broadcast Filter Problem

Certain broadcast filters would cause learning and ARP responses to work incorrectly.

Upgrade Log

Management Memory now contains a history of SCP firmware upgrades, so the version that writes each record can be determined. This revision history can be examined from the OBM Extended Options Menu (SCP revision history).

Hardware Revisions

Only the alphabetic portion of hardware revisions will be reported by OBM and SNMP.

Aging on Hunt Groups

A problem with the aging protocol used over Hunt Groups would occasionally cause a system crash.

Backup Noted in Error Log

An SCP error log entry will note when an SCP becomes a backup, and when it becomes elected.

Fix ifType on STS3c Links

The proper value is now returned for ifType on sonetSTS3c links.

SCP-CLK Synchronization Problem

A synchronization problem between the SCP and CLK cards would occasionally cause the SCP to crash when rebooted.

Dumping the fdbTable

This operation has been speeded up considerably.

AGL PhyType

Previously the SNMP agent would return an error for phyType when no PHY was present.

arpAgent MIB Object

The SNMP agent now can deal with sets of **arpAgent** to invalid values.

Service Class Values

The SNMP agent now returns values of **SapSvcSapValue** even for pre-initialized values. Previously, only management-specified values were returned.

AGL Error Entry Number 800 and Reboot

In certain error recovery scenarios, the AGL-2 or AGL-2+ would log a 800 error and crash.

Known Problems and Restrictions in Version 3.1

Address Lockdown

Addresses that are locked down to a port are not assigned to the proper port when that port's linecard is power cycled.

OAM Disabled for AGL

This MIB object **aglInterfaceATMOAMStatus** is permanently set to "disabled" in this release. It cannot be set to "enabled".

Non Zero VPI

Non zero values of VPI are not supported on the AGL-2 and AGL-2+ modules.

STS3-C vs. STM1 Mode

If the two ends of a SONET link are set to different mode values, the PHY LED on the STM1 side will blink green, indicating that the PHY is down and the link is not functional. In addition, some or all of the following counters will increment on the STM1 side:

aglsonetSectionCurrentESs
aglsonetSectionCurrentSESs
aglsonetSectionCurrentSEFSs

aglsonetLineCurrentESs
aglsonetLineCurrentSESs
aglsonetLineCurrentUASs

aglsonetPathCurrentESs
aglsonetPathCurrentSESs
aglsonetPathCurrentUASs

The STS3-c side of the link may appear to be normal except that the following counters may increment:

- **aglsonetLineCurrentUASs**
- **aglsonetPathCurrentUASs**

Counters

The counters for AAL-5 CRC errors and ATM Header Error Check (HEC) errors are not accessible in this release. The counters for per VCC cells transmitted are not accessible in this release.

UNI 3.0

The current release of AGL does not support the ILMI portion of the UNI 3.0 specification.

AGL-2+ modPHYs

The AGL-2+ supports only T3, E3, and OC-3 modPHYs. Although other modPHYs (T1 and E1) may physically fit in the AGL-2+, they are **not supported** for operation with the GIGAswitch/FDDI.