53-1003138-03
15 September 2014

Fabric OS

FCIP Administrator's Guide

Supporting Fabric OS v7.3.0

BROCADE

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

The authors and Brocade Communications Systems, Inc. assume no liability or responsibility to any person or entity with respect to the accuracy of this document or any loss, cost, liability, or damages arising from the information contained herein or the computer programs that accompany it.

Contents

Preface

Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used in the flow of the text to highlight specific words or phrases.

Format	Description
bold text	Identifies command names Identifies keywords and operands Identifies the names of user-manipulated GUI elements Identifies text to enter at the GUI
italic text	Identifies emphasis Identifies variables and modifiers Identifies paths and Internet addresses Identifies document titles
`Courier font`	Identifies CLI output Identifies command syntax examples

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.

Convention	Description
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, **--show** WWN.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options. In Fibre Channel products, square brackets may be used instead for this purpose.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a “soft” line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

CAUTION

A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER

A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

You can download additional publications supporting your product at www.brocade.com. Select the Brocade Products tab to locate your product, then click the Brocade product name or image to open the individual product page. The user manuals are available in the resources module at the bottom of the page under the Documentation category.

To get up-to-the-minute information on Brocade products and resources, go to MyBrocade. You can register at no cost to obtain a user ID and password.

Release notes are available on MyBrocade under Product Downloads.

White papers, online demonstrations, and data sheets are available through the Brocade website.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online, by telephone, or by e-mail. Brocade OEM customers contact their OEM/Solutions provider.

Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to http://www.brocade.com/services-support/index.html.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone	E-mail
Preferred method of contact for non-urgent issues: • My Cases through MyBrocade • Software downloads and licensing tools • Knowledge Base	Required for Sev 1-Critical and Sev 2-High issues: • Continental US: 1-800-752-8061 • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33) • For areas unable to access toll free number: +1-408-333-6061 • Toll-free numbers are available in many countries.	support@brocade.com Please include: • Problem summary • Serial number • Installation details • Environment description

Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/Solution Provider, contact your OEM/Solution Provider for all of your product support needs.

- OEM/Solution Providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/Solution Provider.

- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.
- For questions regarding service levels and response times, contact your OEM/Solution Provider.

Document feedback

To send feedback and report errors in the documentation you can use the feedback form posted with the document or you can e-mail the documentation team.

Quality is our first concern at Brocade and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

- Through the online feedback form in the HTML documents posted on www.brocade.com.
- By sending your feedback to documentation@brocade.com.

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

About This Document

Supported hardware and software

The following hardware platforms support FCIP as described in this manual:

- Brocade DCX, DCX-4S, DCX 8510-4, and DCX 8510-8 with one or more FX8-24 blades
- Brocade 7800 switch
- Brocade 7840 switch

What is new in this document

Major new additions or deletions in this document support new features related to the Brocade 7840 Extension Switch, additional changes for Fabric OS v7.3.0, and corrections. Major sections of this publication affected by additions and corrections include the following:

- FCIP Concepts and Features
    - Adaptive Rate Limiting - added Enhanced ARL (eARL) information
    - Compression options - added details on the Brocade 7840 switch
    - Open Systems Tape Pipelining - modified "FCIP Fastwrite and OSTP configurations" section
    - Support for IPv6 addressing - added information about PMTU discovery
    - Memory use limitations for large-device tunnel configurations - added details on the Brocade 7840 switch
    - Effect of configuration on tunnel control block memory - added details on the Brocade 7840 switch
    - Firmware downloads - added information on FCIP Hot Code Load (HCL) for the Brocade 7840 switch
- FCIP on Brocade Extension Switches and Blades
    - FCIP platforms and supported features - added details on the Brocade 7840 switch
    - Brocade 7840 Extension Switch - new section
    - Path Maximum Transmission Unit discovery - new section
    - Tunnel and circuit requirements - added new section on Brocade 7840 extension switches and combined requirements for products under Brocade 7840, 7800, and FX8-24 headings
- Configuring FCIP
    - Configuration preparation - modified with Brocade 7840 information
    - Configuring port modes (Brocade 7840 switch) - new section
    - Configuring port speed (Brocade 7840 switch) - new section
    - Configuring an IPIF - modified with Brocade 7840 switch details
    - Configuring an IP route - modified with Brocade 7840 switch details

  - Creating an FCIP tunnel - added section on "Creating Brocade 7840 tunnels" and modified "Tunnel configuration options" sections
  - Configuring FCIP HCL - new section
  - Using FCIP with logical switches - added section on "Brocade 7840 switch considerations and limitations"
  - Managing the VLAN tag table - added information on the Brocade 7840 switch
- FCIP Management and Troubleshooting
  - WAN performance analysis tools - added section "Using WAN Tool"
  - FTRACE configuration - added section on the Brocade 7840 switch
  - Changing configuration settings -- added section "Brocade 7840 switch example"
  - Displaying FTRACE status on an FCIP DP complex - added section "Brocade 7840 switch example"

Updates to version 53-1003138-01 (15 August 2014)

The following changes have been made to version 53-1003138-01:

- The following terms have been changed throughout the publication
  - Enhanced ARL and eARL are now ARL.
  - FCIP HCL is now Extension HCL.
- In Configuring an IP route on page 66, specified that 120 routes are maximum on the 7840 switch.
- In Creating 7840 tunnels on page 69, modified command examples so that committed rates are 20 Gbps or less.
- Modified list of HCL limitations and considerations in Extension HCL limitations and considerations on page 45.
- Revised Using WAN tool on page 119 section. Added "WAN Tool commands," "Configuring a WAN Tool session and displaying results," and "Resolving test session problems" subsections. Added more detail and examples to procedures for configuring WAN Tool sessions.
- Revised description of ARL for the 7840 switch in 7840 Switch support for ARL on page 17.

Updates to version 53-1003138-02 (15 September 2014)

The following changes have been made to version 53-1003138-02:

- In Configuring Extension HCL on page 83, updated command examples for the **portcfg** and **portshow** commands.
- In Configuring backup tunnels on page 84, updated command examples for the **portcfg** and **portshow** commands.
- In Extension Hot Code Load on page 44, removed a statement that Extension HCL runs on both switches at once. Extension HCL processes do not run in parallel.
- In Enabling persistently disabled ports on page 85, corrected the introductory paragraph to match the configuration steps.

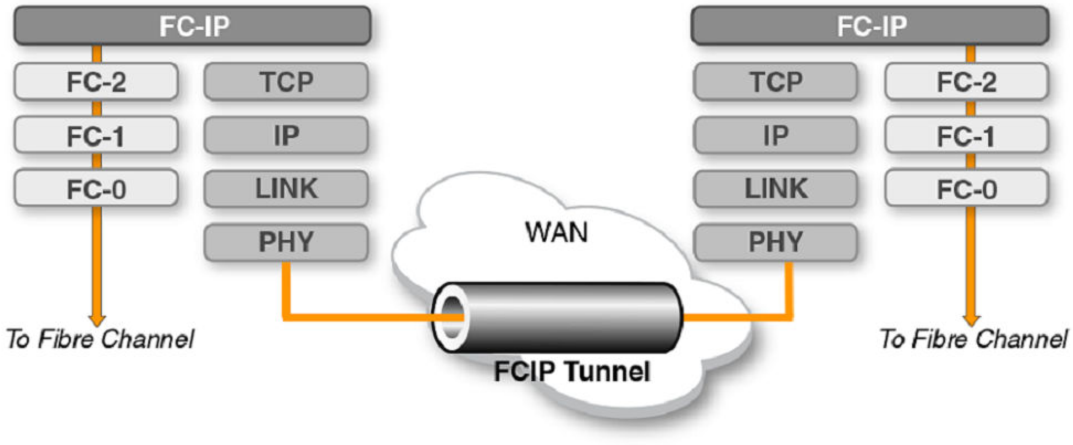
FCIP Concepts and Features

FCIP concepts

Fibre Channel over IP (FCIP) enables you to use the existing IP wide area network (WAN) infrastructure to connect Fibre Channel storage area networks (SANs). FCIP supports applications such as remote data replication (RDR), centralized SAN backup, and data migration over very long distances that are impractical or very costly using native Fibre Channel connections. FCIP tunnels, built on a physical connection between two extension switches or blades, allow Fibre Channel I/O to pass through the IP WAN.

The TCP connections ensure in-order delivery of Fibre Channel (FC) frames and lossless transmission. The Fibre Channel fabric and all Fibre Channel targets and initiators are unaware of the presence of the IP WAN.

FIGURE 1 FCIP tunnel concept and TCP/IP layers



VE_Ports and VEX_Ports

FCIP tunnels emulate FC ports on the extension switch or blade at each end of the tunnel. Once the FCIP tunnels are configured and the TCP connections are established for a complete FCIP circuit, a logical interswitch link (ISL) is activated between the switches. Once the tunnel and ISL or Fibre Channel Routing (FCR) connection are established between the switches, these logical FC ports appear as "virtual" E_Ports or VE_Ports. VE_Ports operate like FC E_Ports for all fabric services and Fabric OS operations. Rather than using FC as the underlying transport VE_Ports use TCP/IP over Ethernet.

A "virtual" FC Port exposed by the FCIP tunnel to form an ISL connection allows you to configure a virtual EX_Port or VEX_Port to support an FCR connection. From the point of view of a switch in an edge fabric, a VEX_Port appears as a normal VE_Port. It follows the same Fibre Channel protocol as other VE_Ports. However, VEX_Ports terminate at the switch and do not allow fabrics to merge by propagating fabric services or routing topology information from one edge fabric to another. This provides fabric isolation on both sides of the IP network.

NOTE
VE_Ports or VEX_Ports cannot connect in parallel to the same domain at the same time as Fibre Channel E_Ports or EX_Ports.

An FCIP tunnel is assigned to a VE_Port or VEX_Port on the switch or blade at each end of the tunnel. Because multiple VE_Ports and VEX_Ports can exist on the extension switch or blade, you can create multiple tunnels through the IP network.

Fibre Channel frames enter FCIP through virtual E_Ports (VE_Ports) or virtual extension ports (VEX_Ports) and are encapsulated and passed to Transmission Control Protocol (TCP) layer connections. An FCIP Data Processing complex (DP complex), on the switch or blade handles the FC frame encapsulation, de-encapsulation, and transmission to the TCP link.

FCIP interfaces, circuits, and trunks

You must configure unique IP interfaces (IPIFs) on each switch or blade Ethernet port used for FCIP traffic. An IPIF consists of an IP address for an Ethernet port, netmask, and an MTU size. For the Brocade 7840 switch, the IPIF can also contain a VLAN ID for VLAN tagging. If the destination FCIP interface is not on the same subnet as the Ethernet port IP address, you must configure an IP route to that destination. You can define a specific number of routes per Ethernet port based on the extension platform. Refer to Tunnel and circuit requirements on page 47 for specifications. A port can contain multiple IP interfaces.

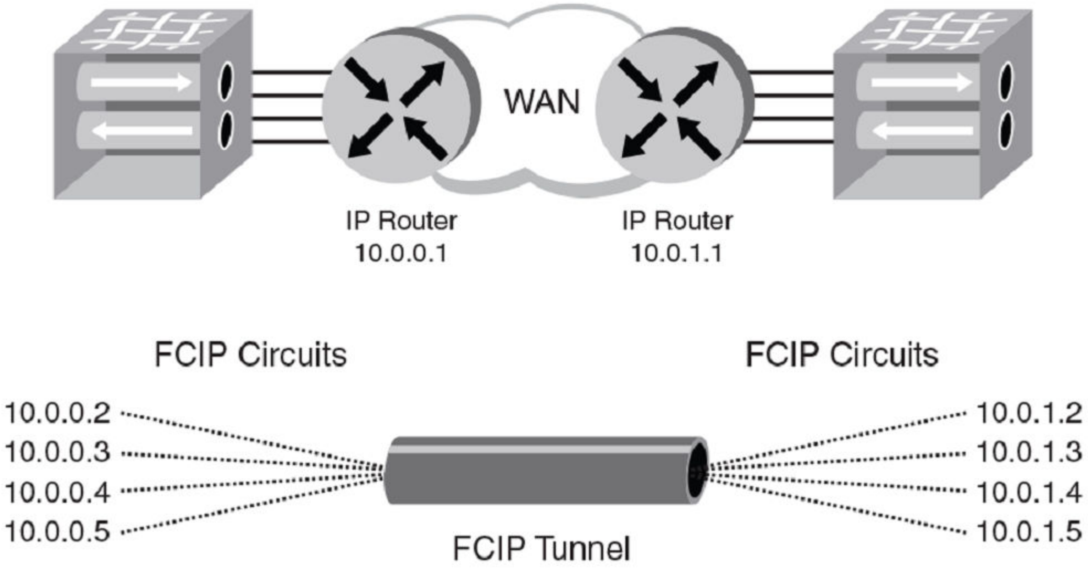
NOTE
In this publication, the "source" is the switch you are configuring, while the "destination" is the switch on the other end of the tunnel.

Configure an FCIP tunnel by specifying a VE_Port for a source and destination interface. When you configure the tunnel, you will provide two IP addresses, one for the source and one for the destination IP interface.

For high availability (HA) tunnels on Brocade 7840 switches, configure two destination IP addresses for the VE_Port on the destination DP complex and two IP addresses for the VE_Port on the source DP complex.

A circuit is a connection between a pair of IP addresses that are associated with source and destination endpoints of an FCIP tunnel. Circuits provide the links for traffic flow between source and destination FCIP interfaces that are located on either end of the tunnel. For each tunnel, you can configure a single circuit or a trunk consisting of multiple circuits. Multiple circuits can be configured per Ethernet port by assigning them unique IPIFs. When you configure a circuit, you provide the IP addresses for its source and destination interfaces.

FIGURE 2 FCIP tunnel and FCIP circuits



An FCIP trunk is a tunnel consisting of multiple FCIP circuits.

For specifications and restrictions on FCIP tunnels, circuits, and trunks for the Brocade 7800 Extension Switch, 7840 Extension Switch, and the FX8-24 Extension Blade, refer to the *FCIP on Brocade Extension Switches and Blades* chapter.

FCIP Trunking

FCIP Trunking is a method for managing the use of WAN bandwidth and providing redundant paths over the WAN that can protect against transmission loss due to WAN failure. FCIP Trunking also provides granular load balancing on a weighted round-robin basis per batch. Trunking is enabled by creating logical circuits within an FCIP tunnel so that the tunnel utilizes multiple circuits to carry traffic between multiple source and destination addresses. For circuit capacities for Brocade extension switches and blades, refer to Tunnel and circuit requirements on page 47.

Redundancy and fault tolerance

Multiple FCIP tunnels can be defined between pairs of extension switches or blades, but doing so defeats the benefits of a multiple-circuit FCIP tunnel. Defining two tunnels between a pair of switches or blades is not as redundant or fault-tolerant as having multiple circuits in one tunnel.

FCIP Trunking provides lossless link loss (LLL). LLL ensures all data lost in flight is retransmitted and placed back in order prior to being delivered to upper layer protocols. This is an essential feature to prevent interface control checks (IFCCs) on mainframes using FICON and SCSI timeouts for open-system-based replication. For more information about LL on specific Brocade extension switches and blades, refer to Circuit failover on page 51.

Tunnel restrictions for FCP and FICON acceleration

Multiple FCIP tunnels are not supported between pairs of extension switches or blades when any of the FICON emulation or acceleration features or Fibre Channel Protocol (FCP) acceleration features are enabled on the tunnel, unless Traffic Isolation (TI) Zones or logical switch/logical fabric (LS/LF) configurations are used to provide deterministic flows between the switches. These features require deterministic FC frame routing between all initiators and devices over multiple tunnels. Non-controlled, parallel (equal-cost multi-path) tunnels are not supported between the switch pairs when emulation is enabled on any one or more tunnels without controlling the routing of SID/DID pairs to individual tunnels using TI Zones or Virtual Fabrics (VF) LS/LF configurations.

Note these additional restrictions:

- FICON networks with FCIP emulating and nonemulating tunnels do not support exchange-based routing (EBR) configurations.
- If one end of a FICON emulating tunnel runs Fabric OS v7.0.0 or later, both ends of the tunnel must run Fabric OS v7.0.0 or later.
- When planning Fabric OS upgrades or downgrades, it is recommended that you upgrade or downgrade both endpoints of the FCIP tunnel (FICON or FCP) concurrently with the same Fabric OS version if either of the FICON or FCP emulation features is active. This is because limited interoperability testing is performed in FICON or FCP emulation configurations between Fabric OS releases.
- When configuring tunnels to support large numbers of devices, consider memory limitations of the Brocade extension switch or blade if you are enabling any type of emulation feature, such as FCP or FICON. If too many devices are present or activated at one time, emulation operations can be negatively impacted. Refer to Memory use limitations for large-device tunnel configurations on page 23.

IP WAN network considerations

Because FCIP uses TCP connections over an existing wide area network, consult with the WAN carrier and IP network administrator to ensure that the network hardware and software equipment operating in the data path can properly support the TCP connections. Keep the following considerations in mind:

- Routers and firewalls that are in the data path must be configured to pass FCIP traffic through a specific TCP port on the switch. The Brocade 7800 and FX8-24 use TCP port 3225 and the Brocade 7840 switch uses port 3225 and 3226. IPsec traffic also passes through the TCP port, if IPsec is used (UDP port 500). The Brocade TCP implementation selects a port between 49152 and 65535 as the ephemeral (or initiating) port to open up to port 3225 and 3226.
- To enable recovery from a WAN failure or outage, be sure that diverse, redundant network paths are available across the WAN.
- Be sure the underlying WAN infrastructure is capable of supporting the redundancy and performance expected in your implementation.

Adaptive Rate Limiting

Adaptive Rate Limiting (ARL) is performed on FCIP circuits to change the rate in which the FCIP tunnel transmits data through the IP network. ARL uses information from the TCP connections to determine and adjust the rate limit for the FCIP circuit dynamically. This allows FCIP connections to utilize the maximum available bandwidth while providing a minimum bandwidth guarantee. ARL is configured on a per-circuit basis because each circuit may have available different amounts of bandwidth.

ARL is supported only if Fabric OS v7.0.0 and later is running on both ends of the FCIP tunnel. For Fabric OS v7.0.0 and later, you can configure minimum and maximum rates for each circuit of a tunnel using the following ports:

- XGE (10 GbE) ports on the Brocade FX8-24 blade (xge0 and xge1).
- 10G (1/10 GbE) ports on the Brocade 7840 switch (ge2-ge17).
- 40 GbE ports on the Brocade 7840 switch (ge0 and ge1).
- 1 GbE ports on the Brocade 7800 switch (ge0-ge5) and Brocade FX8-24 blade (ge0-ge9).

ARL applies a minimum and maximum traffic rate, and allows the traffic demand and WAN connection quality to determine the rate dynamically. If traffic is flowing error-free over the WAN, the rate grows towards the maximum rate. If TCP reports an increase in retransmissions, the rate reduces towards the minimum. ARL never attempts to exceed the maximum configured value and reserves at least the minimum configured value.

For ARL limitations and features specific to supported products, refer to the *FCIP on Brocade Extension Switches and Blades* chapter.

7840 Switch support

ARL on the 7840 switch has been enhanced to react ten times faster to varying traffic patterns that compete for WAN bandwidth or use shared interfaces.

Sharing bandwidth among many applications is common in today's cost conscious enterprises environment. Although ARL has always been able to accommodate shared bandwidth, the storage data crossing FCIP connections has significantly grown and is now consuming larger and faster links. ARL's quicker response time provides faster rate limiting adaptation, permitting optimized throughput of not only FCIP, but also the competing flows.

ARL's back-off mechanism has been optimized to increase overall throughput. In the past, the back-off mechanism was an all or nothing decision. The new intelligence built into ARL allows the rate limiting to back-off in precise steps to preserve throughput. Experience has shown that on a shared link a complete reset back to the floor value is often not required. Preserving bandwidth and reevaluating if additional back-offs steps are required is prudent.

ARL maintains Round Trip Time (RTT) stateful information to better predict network conditions and to allow more intelligent and granular decisions about proper adaptive rate limiting. When ARL encounters a network error, it looks back at pertinent and more meaningful stateful information, which will be different relative to the current state. Rate limit decisions are fine-tuned using this new algorithm.

FSPF link cost calculation when ARL is used

Fabric Shortest Path First (FSPF) is a link state path selection protocol that directs traffic along the shortest path between the source and destination based upon the link cost. When ARL is used, the link cost is equal to the sum of the maximum traffic rates of all established, currently active low metric circuits in the tunnel. The following formulas are used:

- If the bandwidth is greater than or equal to 2 Gbps, the link cost is 500.
- If the bandwidth is less than 2 Gbps, but greater than or equal to 1 Gbps, the link cost is 1,000,000 divided by the bandwidth in Mbps.
- If the bandwidth is less than 1 Gbps, the link cost is 2000 minus the bandwidth in Mbps.

Configuring ARL

To configure the minimum and maximum committed rates for ARL on a circuit, refer to Tunnel configuration options on page 70.

ARL configuration limitations

Consider the following limitations when configuring ARL:

- As a best practice, the aggregate of the circuit minimum rate bandwidth settings through a VE_Port tunnel should not exceed the bandwidth of the WAN link. For example, if the WAN link is 500 Mbps, the aggregate of the ARL maximum rates connected to that WAN link can be no more than 500 Mbps. For ingress rates, there is no limit because the FC flow control (BBC) rate limits the incoming data.
- The aggregate of the minimum configured values cannot exceed the speed of the Ethernet interface, which is 1 Gbps for GbE ports or 10 Gbps for 10 GbE ports, or 40 Gbps for 40 GbE ports.
- Configure minimum rates of all the tunnels so that the combined rate does not exceed the specifications listed for the extension product in the Tunnel and circuit requirements on page 47.
- For 1 GbE, 10 GbE, and 40 GbE ports, the ratio between the minimum committed rate and the maximum committed rate for a single circuit cannot exceed five times the minimum. For example, if the minimum is set to 2 Gbps, the maximum for that circuit cannot exceed 10 Gbps. This is enforced in software.
- The ratio between any two circuits on the same tunnel should not exceed four times the lower circuit. For example, if one circuit is configured to 1 Gbps, any other circuit in that same tunnel should not exceed 4 Gbps. This is *not* enforced in software, but is strongly recommended.

Compression options

Compression options are defined in the **portcfg fciptunnel** *port* **create** and **portcfg fciptunnel** *port* **modify** commands. There are different options for different extension products.

NOTE
Fibre Channel throughput for all compression modes depends on the compression ratio achievable for the data pattern. Brocade makes no promises, guarantees, or assumptions about compression ratio that any application may achieve.

Brocade 7800 switch and FX8-24 blade

The following compression options are available for the Brocade 7800 switch and FX8-24 blade:

- Standard - A hardware compression mode.
- Moderate - A combination of hardware and software compression that provides more compression than hardware compression alone. This option supports up to 8 Gbps of FC traffic.
- Aggressive - Software-only compression that provides a more aggressive algorithm than used for the standard and moderate options. This option supports up to 2.5 Gbps of FC traffic.
- Auto - Allows the system to set the best compression mode based on the tunnel’s configured bandwidth and the aggregate bandwidth of all tunnels in the extension switch or blade.

Follow the guidelines for assigning explicit compression levels for tunnels in the following table.

TABLE 1 Assigning compression levels

Total effective tunnels FC side	Compression level
Equal to or less than 512 Mbps	Aggressive
More than 512 Mbps and less than or equal to 2 Gbps	Moderate
More than 2 Gbps	Standard

Brocade 7840 switch

The following compression options are available for the Brocade 7840 switch.

- None - no compression.
- Deflate - Processor-based compression. This option initiates the processor compression engine in deflate mode with preference on speed. This provides a lower speed than fast-deflate, but more than aggressive deflate (16 Gbps total per DP before compression). The compression is better than fast deflate, but is typically not as good as aggressive deflate.
- Aggressive deflate - Processor-based compression. Initiates the processor engine in deflate mode with preference on compression. This is the slowest (10 Gbps before compression), but typically provides the best compression.
- Fast deflate - Hardware-based compression. This option initiates a deflate-based algorithm to compress data before it enters the DP and decompresses the data after it leaves the DP (egress). This is the highest throughput mode of compression (40 Gbps per DP before compression), but provides the least amount of compression.

Follow the guidelines for assigning explicit compression levels for tunnels in the following table.

TABLE 2 Assigning compression levels

Total tunnel bandwidth on a DP	Compression level
2 Gbps or less	Aggressive deflate
2 Gbps to 4 Gbps	Deflate
More than 4 Gbps	Fast deflate

FastWrite and Open Systems Tape Pipelining

Brocade FastWrite is an algorithm that reduces the number of round trips required to complete a SCSI write operation. FastWrite can maintain throughput levels over links that have significant latency. The RDR (Remote Data Replication) application still experiences latency; however, reduced throughput due to that latency is minimized for asynchronous applications, and response time is cut in half for synchronous applications.

Open Systems Tape Pipelining (OSTP) can be used to enhance open systems SCSI tape write I/O performance. When the FCIP link is the part of the network with the longest latency, OSTP can provide accelerated speeds for tape read and write I/O over FCIP tunnels. To use OSTP, you must enable FCIP Fastwrite also.

OSTP accelerates SCSI read and write inputs and outputs to sequential devices (such as tape drives) over FCIP, which reduces the number of round-trip times needed to complete the I/O over the IP network and speeds up the process.

Both sides of an FCIP tunnel must have matching configurations for OSTP and FCIP FastWrite features to work. FCIP FastWrite and OSTP are enabled by turning them on during the tunnel configuration process. They are enabled on a per-FCIP tunnel basis.

FCIP FastWrite and OSTP configurations

The FCP features used in FCIP FastWrite and OSTP require a deterministic FC Frame path between initiators and targets when multiple tunnels exist. If there are non-controlled parallel (ECMP) tunnels between the same SID/DID pairs, protocol optimization will fail when a command is routed over one tunnel and the response is returned over a different tunnel. To help understand the supported configurations, consider the configurations shown in the following two figures. In both cases, there are no multiple ECMP paths. In the first figure, there is a single tunnel with FastWrite and OSTP enabled. In the second figure, there are multiple tunnels, but none of them create a multiple ECMP path.

Brocade extension devices have the intelligence to distinguish between storage flows that use protocol optimization and those that do not use protocol optimization. For example, IBM SVC does not use FastWrite, but EMC SRDF/A does use FastWrite. Both applications functioning over the FCIP connection are fully supported for FastWrite because FastWrite will not engage with the IBM SVC flows while still engaging with the SRDF/A flows across the same FCIP VE_Port. This is also true when using OSTP with IBM SVC. Both flows can utilize the same VE_Port with FastWrite and OSTP enabled. The IBM SVC will not engage the protocol optimization.

FIGURE 3 Single tunnel, FastWrite and OSTP enabled

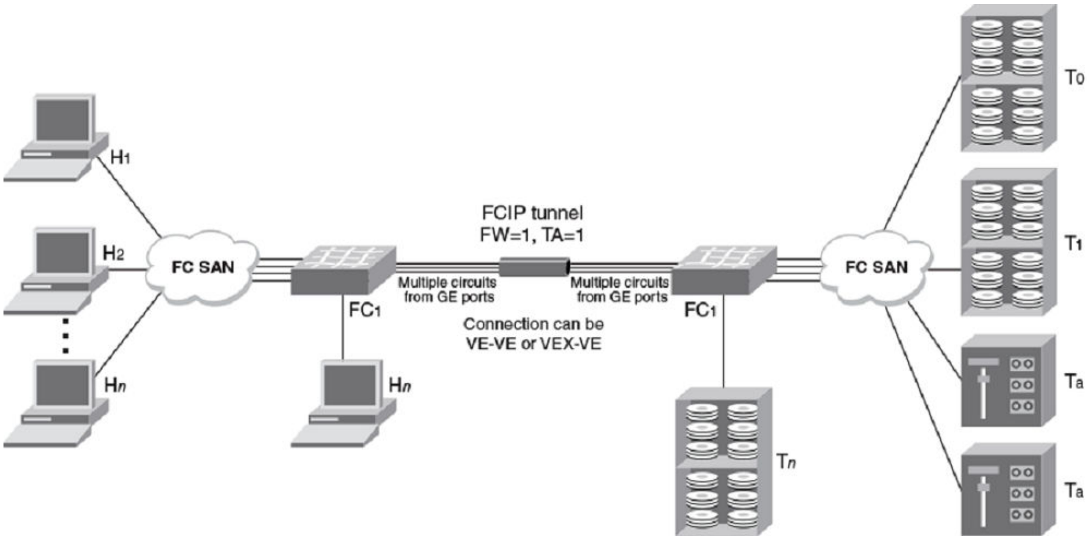
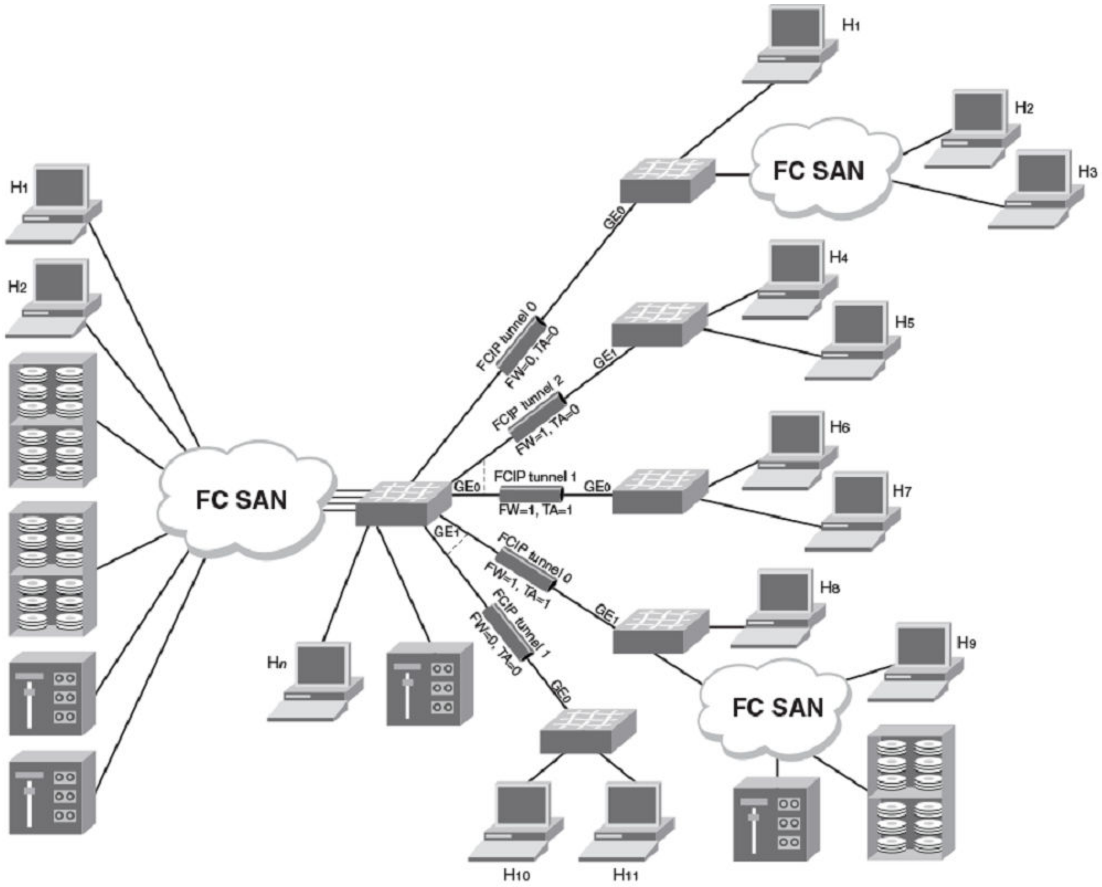

H1
H2
Hn
FC SAN
FC1
Hn
FCIP tunnel
FW=1, TA=1
Multiple circuits from GE ports
Multiple circuits from GE ports
Connection can be VE-VE or VEX-VE
FC1
FC SAN
Tn
T0
T1
Ta
Ta

FIGURE 4 Multiple tunnels to multiple ports, FastWrite and OSTP enabled on a per-tunnel, per-port basis



In some cases, Traffic Isolation Zoning or VF LS/LF configurations may be used to control the routing of SID/DID pairs to individual tunnels. This provides deterministic flows between the switches and allows the use of ECMP. Refer to the *Fabric OS Administrator's Guide* for more information about Traffic Isolation Zoning.

Support for IPv6 addressing

The IPv6 implementation is a dual-IP layer operation implementation as described in RFC 4213. IPv6 addresses can exist with IPv4 addresses on the same interface, but the FCIP circuits must be configured as IPv6-to-IPv6 and IPv4-to-IPv4 connections. IPv6-to-IPv4 connections are not supported. Likewise, encapsulation of IPv4 in IPv6 and IPv6 in IPv4 is not supported.

This implementation of IPv6 uses unicast addresses for the interfaces with FCIP circuits. Unicast addresses must follow the RFC 4291 IPv6 standard. This IPv6 implementation uses the IANA-assigned IPv6 Global Unicast address space (2000::/3). The starting three bits must be 001 (binary) unless IPv6 with embedded IPv4 addresses is used. The link-local unicast address is automatically configured on the interface, but using the link-local address space for FCIP circuit endpoints is not allowed. Site-local unicast addresses are not allowed as FCIP circuit endpoints.

Note the following IPv6 addressing points:

- Anycast addresses are not used. Each IPv6 interface has a unique unicast address and addresses configured are assumed to be unicast.
- Multicast addresses cannot be configured for an IPv6 interface with FCIP circuits. The IPv6 interface does not belong to any Multicast groups other than the All-Nodes Multicast and the Solicited-Node Multicast groups (these do not require user configuration).
- The IPv6 implementation follows the RFC 2460 standard for the 40-byte IPv6 header format.
- The IPv6 8-bit Traffic class field is defined by the configured Differentiated Services field for IPv6 (RFC 2474). The configuration of this is done on the FCIP circuit using the Differentiated Services Code Point (DSCP) parameters to fill the 6-bit DSCP field.
- Flow labels are not supported on this IPv6 implementation. The 20-bit Flow Label field defaults to all zeros.
- The IPv6 optional Extension Headers are not supported. The optional Extension Headers (zeros) inserted into any ingress packets that contain these headers will be discarded. The next header field must be the Layer 4 protocol for this implementation.
- Parts of the Neighbor Discovery protocol (RFC 4861) are used in this implementation.
    - Hop limits (such as Time to Live (TTL)) are learned from the Neighbor Advertisement packet.
    - The link-local addresses of neighbors are learned from Neighbor Advertisement.
    - The netmask is deprecated in IPv6. Instead, the prefix length notation is used to denote subnets in IPv6 (the Classless Inter-Domain Routing (CIDR) addressing syntax). Prefix length of neighbor nodes is learned from the received Neighbor Advertisement packet.
    - The IPv6 link-local address for each GE interface is configured at startup and advertised to neighbors. The user does not configure the interface link-local address.
- The 8-bit hop limit field is filled by the learned value during Neighbor Discovery.
- IPv6 addresses and routes must be statically configured by the user. Router Advertisements and IPv6 Stateless Address Autoconfiguration (RFC 2462) are not supported.
- The Neighbor Discovery ICMPv6 Solicitations and Advertisements are transmitted to the Layer 2 Ethernet multicast MAC address derived from the IPv6 source address (RFC 2464).
- ICMPv6 message types in RFC 4443 and ICMPv6 message types used for Neighbor Discovery are supported.
- Path MTU Discovery
    - For the Brocade 7800 switch and FX8-24 blade, Path MTU (PMTU) discovery is not supported. The MTU option in the **portcfg ipif** command is optional. If not configured, 1500 bytes is used. The maximum IP MTU supported is 1500 bytes (including the 40-byte fixed IPv6 header), the same as for IPv4. The minimum IP MTU allowed is 1280 bytes, including the 40-byte fixed IPv6 header. Any network used for IPv6 FCIP circuits must support an IP MTU of 1280 bytes or larger. IPv6 fragmentation is not supported. The Layer 4 protocol ensures that the PDU is less than the IP MTU (including headers).
    - For the Brocade 7840 switch, PMTU discovery is supported. Refer to Path Maximum Transmission Unit discovery on page 47.
- IPv6 addressing with IPsec:
    - For the Brocade 7840 switch, IPv6 addressing can be used when implementing IPsec.
    - For the Brocade 7800 switch and FX8-24 blade, IPv6 addressing cannot be used when implementing IPsec.

IPv6 with embedded IPv4 addresses

Only IPv4-compatible IPv6 addresses are supported. Only the low-order 32 bits of the address can be used as an IPv4 address (the high-order 96 bits must be all zeros). This allows IPv6 addresses to be used on an IPv4 routing infrastructure that supports IPv6 tunneling over the network. Both endpoints of the circuit must be configured with IPv4-compatible IPv6 addresses. IPv4-to-IPv6 connections are not supported. IPv4-mapped IPv6 addresses are not supported because they are intended for nodes that support IPv4 only when mapped to an IPv6 node.

Memory use limitations for large-device tunnel configurations

The FCIP data processing layer on the Brocade extension switch and blade data processing (DP) complex has access to reserved memory used for control block structure allocations. Following are related specifications for the Brocade switches and blades.

TABLE 3 VE_Ports and DRAM pool sizes for extension products

Product	DP VE_Ports	DP DRAM pool size
Brocade 7800	DP0 - 16 through 23	200 Mb
Brocade 7840	DP0 - 24 through 33	512 Mb
	DP1 - 34 through 43	512 Mb
Brocade FX8-24	DP0 - 22 through 31	268 Mb
	DP1 - 12 through 21	268 Mb

Use the **portshow xtun** *slot/ve* **-dram2** command to display current consumption of the FCIP tunnel DP complex control block memory pool. Following is an example of command use with the portion of output showing total DRAM2 pool size and current consumption for a Brocade 7840 switch.

```
switch43:root> portshow xtun 25 -dram2
Dram2 Pool Info:
---------------------------------------------
Total Bytes in DRAM2 Pool: 1336910592 (free) 1504640 (fastfreed)
Total DRAM Bytes Allocated:    6771328 (in use)
```

FCIP tunnel processing will create more control blocks when any type of emulation feature is enabled, such as FCP or FICON. In those cases, be sure to not include too many "devices' in the FCIP tunnel configuration. If too many devices are present or activated at one time, emulation operations can be negatively impacted. Even without emulation enabled, too many devices in the FCIP tunnel may impact operations at some point because of memory consumption. Note that a configuration that works without an emulation feature, such as FICON Acceleration and Fastwrite or Open Systems Tape Pipelining (OSTP), may not work when emulation features are enabled.

Control blocks created during FCP traffic flow

For FCP traffic flows, FCIP tunnel processing creates control block structures based upon the SID/DID pairs used between initiators and devices. If either Fastwrite or OSTP (read or write) is enabled, additional structures and control blocks are created for each logical unit number (LUN) on a SID/DID-pair basis. FCP processing in an emulated tunnel configuration will create multiple control blocks for each LUN if there are multiple SID/DID pairs that can be used to access that same LUN. Each FCP-identified SID/DID flow will be recorded in a structure called an ITN (initiator, target, nexus). Each specific LUN on a SID/DID flow will have an ITL (initiator, target, LUN) control block created for the flow. FCIP FCP emulation processing also creates a structure for each outstanding FC exchange called a turbo write block (TWB).

Control blocks created during FICON traffic flow

For FICON traffic flows, FCIP tunnel processing creates a control block structure based upon the SID/DID pairs called a FICON device path block (FDPB). If any FICON emulation feature is enabled, additional control blocks are created for each SID/DID pair, logical partition (LPAR) number (FICON channel block structure), LCU number (FICON control unit block structure), and for each individual FICON device address on those LCUs (FICON device control block structure).

The total number of FICON device control blocks that will be created over a FICON emulating tunnel is represented by the following equation:

FDCBs = Host Ports x Device Ports x LPARs x LCUs x FICON Devices per LCU

This number grows quickly in extended direct-attached storage device (DASD) configurations, such as those used in IBM z/OS Global Mirror/XRC configurations.

FDCBs example

Assume that the tunnel is used to extend two channel path identifiers (CHPIDs) from a System Data Mover (SDM) site to a production site. Assume also that there are two SDM-extended LPARs and that the IBM DS8000 production controllers have 32 LCUs per chassis, and each LCU has 256 device addresses.

Using the preceding equation, the number of extended FICON device control block images created would be the following:

2 * 2 * 1 * 32 * 256 = 32,768

Effect of configuration on tunnel control block memory

The **portshow xtun** *slot/ve-port* **-fcp -port -stats** command displays current usage and control block sizes per tunnel once control blocks have been allocated. Use output from this command to determine the unique characteristics for a specific tunnel configuration. The highlighted text in the following example shows statistics for the control block structures created for FCP and FICON traffic flows during FCIP tunnel processing.

```
portshow xtun 1/12 -fcp -port -stats
Slot(1) VePort(12) Port Stats: OK
  Global Queue Stats:
   Name,cnt,max,usage,size,total size
   Data,12,14,44,8192,98304
   Message,0,1,9,1432,0
   Stat,40,40,80,928,37120
   Stat Cache,0,40,40,0,0
   Global stats,0,0,0,0,153240
  Port Queue Stats:
   Name,cnt,max,usage,size,total size
   Image,40,40,80,0,0
   SRB,0,0,0,0,0
   TWB,1,40,356,0,0
  Port Struct Allocation Stats:
   Name,cnt,max,usage,size
   IMAGE,40,40,80,3560
   ITN,26,0,0,2984
   ITL,1103,0,0,2312
   FDPB,30,0,0,3864
   FCHB,108,0,0,1400
   FCUB,1721,0,0,1592
   FDCB,14738,0,0,920
  Global Buffer Stats:
   Name,current,min,max
   Write Data Storage,0,0,0
   Read Data Storage,0,0,0
```

```
XBAR % avail,98,98,100
WIRE % avail,95,95,100
```

Use output from **portshow xtun** *slot/ve-port* **-fcp -port -stats** command in conjunction with output from the **portshow xtun** *slot/ve-port* **-dram2** command to determine how a tunnel configuration is affecting FCIP tunnel control block memory. As a rule of thumb, no more than 80 percent of the FCIP tunnel DP complex control block memory pool (dram2) should be allocated for SID/DID pair-related control blocks (ITNs, ITLs, FDPBs, FCHBs, FCUBs, and FDCBs). When more than 80 percent of the pool is allocated, the FCIP tunnel configuration should be redesigned to ensure continuous operation. The design should include examining the existing number of SID/DID pairs in the configuration and determining whether new switches, chassis, or blades should be acquired to reduce the percentage of current usage of the DRAM2.

For Fabric OS v7.2.0 and later, RASlog message XTUN-1008 provides notification of DRAM2 memory usage. The message is generated by the FCIP DP complex when significant memory thresholds are reached. The following thresholds are shown for the Brocade 7800 switch and FX8-24 blade.

The Brocade 7800 generates the XTUN-1008 RASlog message when the following percentages of memory pool are available:

- 66%
- 33%
- 17%
- 8%
- .07%

Each FX8-24 blade and Brocade 7840 FCIP DP complex generates the XTUN-1008 RASlog message when the following percentages of memory pool are available:

- 50%
- 25%
- 12.5%
- 6.25%
- .05%

The RASlog message contents include the amount of allocated memory from the pool, the amount of free memory in the pool, and the total pool size. Use the RASlog message contents to determine if you need to reduce the size of the extended configuration or to plan for additional FCIP switch resources.

Brocade switches and blade DPs are expected to support no more than the number of FICON device control blocks (FDCBs) and extended LUNs (ITLs) noted in the following table.

TABLE 4 FDCBs and ITLs per product DP

Product	FDCB	ITL
Brocade 7800	120,000	50,000
Brocade 7840	512,000	200,000
Brocade FX8-24	160,000	65,000

The Brocade 7840 switch has 1.3 GB of DRAM2 memory allocated per DP. During FCIP hot code load (HCL) operations, duplicated emulation control blocks are created on the same DP for the high-availability portion of the tunnel. That means that at one point in time during the Extension HCL process, twice the normal memory requirements are consumed. This duplication process occurs on the remote non-Extension HCL DP when the primary local DP is going through feature disable processing.

The amount of DRAM2 memory on the Brocade 7840 should be able to support Extension HCL operations with approximately 512K FICON devices active through the VE_Ports on that DP.

Because each customer configuration is unique, the supported number and types of devices will be different. In large configurations, the FCIP administrator should review memory usage periodically to ensure continued, reliable operations of the FCIP tunnel and emulation features.

Firmware downloads

For the Brocade 7800 switch and FX8-24 blade, if FCIP Fibre Channel traffic is active on Fibre Channel ports, the traffic will be impacted during a firmware download.

The Brocade 7840 switch supports the FCIP hot code load (HCL) feature. During an Extension HCL, traffic is failed over to one DP complex as firmware upgrades in the other DP complex. With Extension HCL, active traffic on Fibre Channel ports is not impacted during a firmware download. For more information on this process, refer to Extension Hot Code Load on page 44.

For FCIP, the best practice is to always update the switch or blade at both ends of the tunnel with the same maintenance release of Fabric OS.

For details on downloading firmware, refer to the chapter on installing and maintaining firmware in the *Fabric OS Administrator's Guide*.

FCIP on Brocade Extension Switches and Blades

FCIP platforms and supported features

There are three Brocade platforms that support FCIP:

- The Brocade 7800 switch
- The Brocade FX8-24 blade (Brocade DCX, DCX-4S, DCX 8510-8, and DCX 8510-4 chassis)
- The Brocade 7840 switch

Note the following about FCIP connections between these products:

- FCIP connections are not supported between the Brocade 7800 switch or FX8-24 blades and previous generation products such as Brocade 7500 switches or FR4-18i blades.
- FCIP connections are not supported between Brocade 7840 and previous generation products, such as Brocade 7800 switches, Brocade 7500 switches, FX8-24 blades, and FR4-18i blades.

The following table provides details about platform capabilities.

TABLE 5 FCIP capabilities by platform

Capabilities	7800 switch	FX8-24 blade	7840 switch
FCIP Trunking	Yes	Yes	Yes
Adaptive Rate Limiting	Yes	Yes	Yes
10 GbE ports	No	Yes	Yes (1/10 Gbps)
40 GbE ports	No	No	Yes Enabled using 7840 WAN Rate Upgrade 2 license.
FC ports	Yes (1, 2, 4, 8 Gbps)	Yes (1, 2, 4, 8 Gbps)	Yes (2, 4, 8, 16 Gbps)
Compression	Yes LZ (Lempel-Ziv) and Deflate	Yes LZ and Deflate	Yes Deflate, Aggressive Deflate, Fast Deflate

TABLE 5 FCIP capabilities by platform (Continued)

Capabilities	7800 switch	FX8-24 blade	7840 switch
Protocol acceleration • FCIP Fastwrite • Open Systems Tape Pipelining - OSTP read - OSTP write	Yes	Yes	Yes
QoS • Marking DSCP • Marking 802.1P - VLAN tagging • Enforcement 802.1P - VLAN tagging	Yes	Yes	Yes
FICON extension • FICON emulation • IBM z/OS Global Mirror (formerly eXtended Remote Copy or XRC) acceleration • Tape read acceleration • Tape write acceleration • Teradata emulation • Printer emulation	Yes	Yes	Yes
IPsec • AES-256-GCM • SHA-512 HMAC • IKEv2	Yes Transport mode encrypted data transfer (ESP) method	Yes Transport mode encrypted data transfer (ESP) method	Yes Transport mode encrypted data transfer (ESP) method
VEX_Ports	Yes	Yes	No
Support for third-party WAN optimization hardware	Yes Support limited to Silver Peak for Fabric OS v7.1.0b and later and to Riverbed for Fabric OS v6.4.x and v6.4.x only.	Yes[1]	No
IPv6 addresses for FCIP tunnels	Yes[2]	Yes[2]	Yes[2]
Support for jumbo frames	No IP MTU of 1500 is maximum	No IP MTU of 1500 is maximum	Yes IP MTU of 9216 is maximum
Path Maximum Transmission Unit (PMTU) Discovery	No	No	Yes Maximum discoverable size is 9100 bytes.

The following notes apply the preceding table:

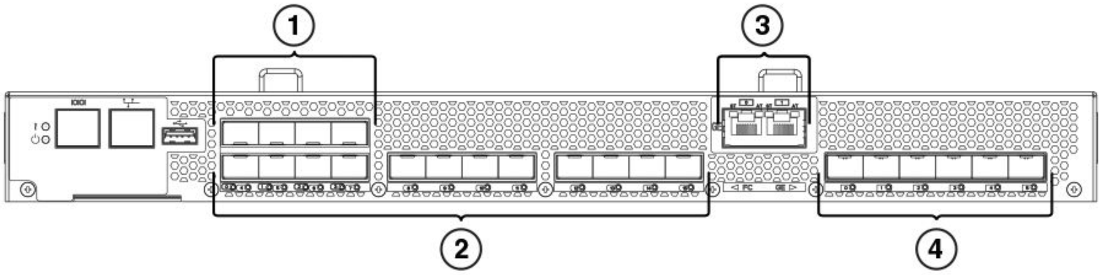
1. Not supported in Fabric OS v7.0 and later.
2. IPv6 addressing is not supported in conjunction with IPsec.

Brocade 7800 Extension Switch

This section provides information on ports, circuits, and tunnels specific to the Brocade 7800 Extension Switch.

The following figure shows the FC ports and GbE ports on the Brocade 7800 switch. There are 16 FC ports, numbered 0 through 15. The FC ports can operate at 1, 2, 4, or 8 Gbps. There are six GbE ports, numbered 0 through 5. Ports 0 and 1 are available as either RJ-45 ports or small form-factor pluggable (SFP) transceiver ports. Only six total GbE ports can be used. The six GbE ports together can provide up to 6 Gbps total bandwidth (full duplex).

FIGURE 5 Brocade 7800 switch FC and GbE ports



1. FC ports 0 through 3
2. FC ports 4 through 15
3. Copper GbE ports 0 and 1 (These ports are RJ-45 copper alternatives for GbE ports 0 and 1.)
4. GbE ports 0 through 5

The Brocade 7800 switch comes in two models:

- The Brocade 7800 4/2 base model uses FC ports 0 through 3, and GbE ports 0 and 1. The GbE ports can be either copper or optical. The RJ-45 copper ports are the default ports. Consider the following when using these ports:
  - Copper ports do not support auto-sense functions.
  - With copper media, auto-negotiation must be enabled on the other end of the port connection.
- The Brocade 7800 16/6 uses FC ports 0 through 15, and GbE ports 0 through 5. The 7800 Upgrade license is required. A 7800 Upgrade license can be purchased for a Brocade 7800 4/2, which enables 12 more Fibre Channel ports for a total of 16, and enables the use of 4 more optical GbE ports for a total of 6.

License options

Important FCIP and FICON extension capabilities of the Brocade 7800 switch require feature licenses. Use the **licenseshow** command to display license keys and licenses currently installed.

NOTE
FCR is not supported on a Brocade 7800 switch that has been partitioned for Virtual Fabrics.

TABLE 6 Brocade 7800 feature licenses

Feature	Purpose	License (licenseShow output)
7800 upgrade	Enables full hardware capabilities, full FCIP tunnel capabilities, and support of Open Systems Tape Pipelining (OSTP). **NOTE** You must reboot the switch to activate this license.	7800 Upgrade license
Advanced FICON acceleration	Enables accelerated tape read/write and IBM z/OS Global Mirror, Teradata, and printer emulation features in FICON environments. Slot-based license.	Advanced FICON Acceleration (FTR_AFA) license
Integrated routing (IR)	Required to configure VEX_Ports to support Fibre Channel Routing (FCR). Chassis-based license.	Integrated Routing license
Advanced extension	Required for multiple-circuit tunnels, FCIP Trunking, Adaptive Rate Limiting (ARL), and other FCIP features.	Advanced Extension (FTR_AE) license

For complete information about the licenses described in the preceding table and additional licenses available for the switch, refer to the *Fabric OS Software Licensing Guide*.

FX8-24 Extension Blade

This section provides information on ports, circuits, and tunnels specific to the FX8-24 Extension Blade.

The figure below shows the FC ports, GbE ports, and 10 GbE ports on the FX8-24 blade. There are 12 FC ports, numbered 0 through 11. The FC ports can operate at 1, 2, 4, or 8 Gbps. There are ten GbE ports, numbered 0 through 9. Ports xge0 and xge1 are 10 GbE ports.

The FX8-24 blade supports two FCIP data processor (DP) complexes, sometimes called "FCIP complexes." Each DP complex has a "home" or local 10 GbE XGE interface and controls a specific range of GbE and VE_Ports.

The DP complex with home XGE port 0 (xge0):

- Controls VE_Ports 22 through 31
- Controls GbE ports 0 through 9
- Has a maximum bandwidth of 10 Gbps

The DP complex with home XGE port 1 (xge1):

- Controls VE_Ports 12 through 21.
- Has a maximum bandwidth of 10 Gbps

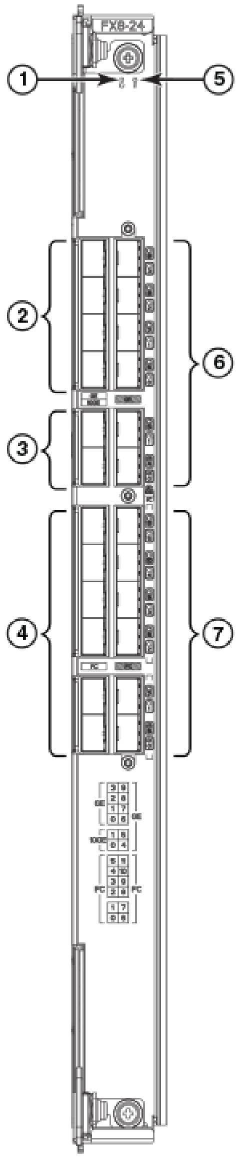
The FX8-24 blade allows a maximum of 20 Gbps full-duplex bandwidth for tunnel connections and can operate in one of three different modes:

- 1 Gbps mode - You can use all ten GbE ports (0 through 9). Both XGE ports are disabled.
- 10 Gbps mode - You can use the xge0 and xge1 ports.
- Dual mode - You can use GbE ports 0 through 9, and port xge0.

The FX8-24 blade can be deployed in a Brocade DCX, DCX-4S, DCX 8510-8, or DCX 8510-4 chassis.

The following figure illustrates ports and LED indicators on the FX8-24 blade.

FIGURE 6 FX8-24 blade FC and GbE ports



1. Power LED
2. GbE ports 0 through 3
3. 10 GbE ports (Labeled xge0 and xge 1 on the blade.)
4. FC ports 0 through 5
5. Status LED

6. GbE ports 4 through 9
7. FC ports 6 through 11

Removing FX8-24 blades

ATTENTION

If you are permanently removing a blade from a Brocade DCX, DCX-4S, DCX 8510-8, or DCX 8510-4 chassis to relocate to another slot in the chassis or you are removing the blade from the chassis entirely, you must follow these procedures *before removing the blade*.

- Remove all FCIP configuration settings for the blade. If there are residual configuration settings, they may cause issues with future configurations and upgrades.
- Delete the IP addresses assigned to the original slot using the **portcfg ipif delete** command. If this is not done, you must return the FX8-24 blade to the original slot and delete the IP addresses.

License options

Important FCIP and FICON extension capabilities of the FX8-24 blade require the feature licenses shown in the following table. Use the **licenseShow** command to display license keys and licenses currently installed.

TABLE 7 FX8-24 FCIP feature licenses

Feature	Purpose	License (licenseShow output)
10 GbE support	Allows 10 Gbps operation on 10 GbE ports. Slot-based license.	10 Gigabit FCIP/Fibre Channel (FTR_10G) license
Advanced FICON acceleration	Enables accelerated tape read/write and IBM z/OS Global Mirror, Teradata, and printer emulation features in FICON environments. Slot-based license.	Advanced FICON Acceleration (FTR_AFA) license
Integrated routing (IR)	Required to configure VEX_Ports to support Fibre Channel Routing (FCR). Chassis-based license.	Integrated Routing license
Advanced extension	Required for multiple-circuit tunnels, FCIP Trunking, Adaptive Rate Limiting (ARL), and other FCIP features. Slot-based license.	Advanced Extension (FTR_AE) license

For complete information about the licenses described in the preceding table and additional licenses available for the switch, refer to the *Fabric OS Software Licensing Guide*.

10 GbE port considerations

Enhanced 10 GbE port operation is different than GbE port operation and requires special considerations when configuring circuits, tunnels, failover operations, and bandwidth. For a complete list of tunnel, circuit, and IP address requirements and capacities, refer to FX8-24 Extension Blade on page 30.

Multigigabit circuits

For each 10 GbE port on an FX8-24 blade, you can configure multigigabit circuits. For example, a single 10 Gbps circuit or two 5 Gbps circuits can be configured per port. A limit of ten FCIP circuits can be configured on a single port. The blade at each end of the tunnel must be running Fabric OS v7.0 or later if the committed rate for circuits exceeds 1 Gbps. The maximum committed rate for a circuit between 10 GbE ports is 10 Gbps.

NOTE
There is no difference in latency or throughput performance for single or multigigabit circuits.

Crossports

When an FCIP DP complex is not using its local XGE interface (xge0 or xge1), but is using the alternate or remote interface, that interface is known as a "crossport." The crossport for xge0 is xge1 and for xge1, the crossport is xge0. Crossports are only supported in the FX8-24 blade.

Typically, IP addresses (IPIFs) used by ge0 through ge9 and xge1 are used for any FCIP circuits that use VE_Ports 12 through 21. The xge1 port is the local XGE interface for VE_Ports 12 through 21. Likewise, IP addresses configured for xge0 are used by circuits for VE_Ports 22 through 31. The xge0 port is the local XGE interface for VE_Ports 22 through 31).

Configure a crossport by assigning an IP address to the remote XGE port that can be used by the local XGE port. For example, assigning an IP address to xge0 as a crossport makes the address available on the remote xge0 for VE_Ports 12 through 21 on the local xge1.

You can also assign IP routes (iproutes) used by the local port, VLAN tagging, and circuits with metrics to the remote XGE port to allow failover to the crossports.

Crossports contain the IP addresses (IPIFs) and IP routes (iproutes) that belong to the remote interface. To use crossports, both XGE ports must be configured in 10 Gbps mode.

Configuring crossports

Configure crossport XGE port addresses using the **--crossport** or **-x** (shorthand) options for the **portcfg ipif** command, as shown in the following example. Note that in this example, IP address 192.168.11.20 is made available on xge0 for circuits on VE_Ports 12 through 21 on local port xge1.

1. Configure an interface for the local XGE port (xge1).

```
switch:admin> portcfg ipif 8/xge1 create 192.168.10.20 netmask
255.255.255.0 mtu 1500
Operation Succeeded
```

2. Configure interface 192.168.11.20 on remote port xge0 to be available for VE_Ports 12 through 21.

```
switch:admin> portcfg ipif 8/xge0 create 192.168.11.20 netmask 255.255.255.0
mtu 1500 --crossport
```

or

```
switch:admin> portcfg ipif 8/xge0 create 192.168.11.20 netmask 255.255.255.0
mtu 1500 -x
```

The output from **portshow ipif** for xge1 shows the crossport tag.

```
switch43:root>portshow ipif 8/xge1
Port         IP Address            / Pfx  MTU   VLAN  Flags
-------------------------------------------------------------
```

```
 8/xge1      192.168.10.20         / 24   1500  0     U R M
 8/xge1      192.168.11.20         / 24   1500  0     U R M X
```

Delete the crossport address using the delete option instead of the create option for the **portcfg ipif** command.

```
switch43:root>portcfg ipif 8/xge1 delete 192.168.11.20 netmask 255.255.255.0
mtu 1500 -x
```

When deleted, output from **portshow ipif** for xge1 will not show the crossport.

```
switch43:root>portshow ipif 8/xge1
Port         IP Address            / Pfx  MTU   VLAN  Flags
-------------------------------------------------------------
 8/xge1      192.168.10.20         / 24   1500  0     U R M
```

NOTE
If the **crossport** or **-x** option is not specified and the address is on the crossport, the command will fail with an unknown IP address. The command will also fail if the crossport option is specified and the address is not on the crossport.

Display local and crossport interface configuration details for a specific XGE port using the **portshow ipif** *slot/xgeport* command. Use the **portshow ipif** command to display details for all interfaces.

```
portshow ipif 8/xge0
portshow ipif
```

Configuring 10 GbE lossless failover with crossports

Refer to 10 GbE Lossless Link Loss (FX8-24 blade) on page 56.

Configuring IP routes with crossports

You can configure IP routes with crossport addresses using the **portcfg iproute** *[slot\port]* **create** command, as in the following example. In the example, the route will be available for FCIP tunnel circuits using VE ports 12 through 21.

```
portcfg iproute 8/xge0 create 1.1.1.0 netmask 255.255.255.0 192.168.11.250 --
crossport
```

or

```
portcfg iproute 8/xge0 create 1.1.1.0 netmask 255.255.255.0 192.168.11.250 -x
```

Delete the route using the **deletedelete** option instead of the **create** option for the **portcfg iproute** command.

```
portcfg iproute 8/xge0 delete 1.1.1.0 netmask 255.255.255.0 -x
```

NOTE
If the **crossport** or **-x** option is not specified and the address is on the crossport, the command will fail with an unknown IP address. The command will also fail if the crossport option is specified and the address is not on the crossport.

Display the static IP routes for the local interface and crossport using the **portshow iproute** command:

```
portshow iproute 1/xge0
```

Display the IP interface configured for he local interface and crossport using the **portshow ipif** command.

```
 portshow ipif 1/xge0
```

For more information on configuring an IP route, refer to Configuring an IP route on page 66.

NOTE
If an XGE port has both regular and crossport addresses configured on it, and they use the same IP route, then two routes must be configured: a regular route and an identical route on the crossport.

Configuring VLAN tags with crossports

Add entries with crossport addresses to the VLAN tag table using the **portcfg vlantag** *[slot/port]* **add** command, as in the following example. This example allows VE ports 12 through 21 to use the configured local IP interface with this VLAN tag.

```
portcfg vlantag 8/xge0 add 192.168.11.20 200 1 --crossport
```

or

```
portcfg vlantag 8/xge0 add 192.168.11.20 200 1 -x
```

Delete the VLAN tag using the **delete** option instead of the **add** option for the **portcfg vlantag** command.

```
portcfg vlantag 8/xge0 delete 192.168.11.20 200 1 -x
```

Display VLAN tag configuration using the **portshow vlantag** command.

NOTE
To tag Class F traffic or data path traffic, use the **-v** or **- -vlan-tagging** option for the **fcipcircuit create** or **fcipcircuit modify** command. The **portcfg vlantag** command is primarily used for ping and traceroute operation and not for tunnels and circuits.

For more information on managing VLAN tags, refer to Managing the VLAN tag table on page 105.

Displaying VLAN tag configuration using the portshow vlantag command

Following is an example for displaying VLAN tagging information for port 0 on blade 8.

```
portshow vlantag 8/xge0
```

For more information on managing VLAN tags, refer to Managing the VLAN tag table on page 105.

For more information on using Fabric OS commands, optional arguments, and command output refer to the *Fabric OS Command Reference*.

Using ping with crossports

You can ping crossport addresses, as in the following example. Note that if the crossport or x options are not specified and the address is on the crossport, the **portCmd** command will fail with an unknown IP address.

```
portcmd --ping 8/xge0 -s 192.168.11.20 -d 1.1.1.1 --crossport
```

or

```
portcmd --ping 8/xge0 -s 192.168.11.20 -d 1.1.1.1 -x
```

When using VLANS, VLAN tagging ensures that test traffic traverses the same path as real FCIP traffic. A VLAN tag entry for both the local and remote sides of the route must exist prior to using the **portCmd --ping** command. Refer to Managing the VLAN tag table on page 105 for details.

For more information on using ping, refer to Using ping to test a connection on page 118.

Using traceroute with crossports

You can trace a route to a crossport address, as in the following example. Note that if the crossport or x options are not specified and the address is on the crossport, the **portCmd** command will fail with an unknown IP address. The command will also fail if the -x option is specified and the address is not on the crossport.

```
portcmd --traceroute 8/xge0 -s 192.168.11.20 -d 1.1.1.1 --crossport
```

or

```
portcmd --traceroute 8/xge0 -s 192.168.11.20 -d 1.1.1.1 -x
```

When using VLANS, VLAN tagging ensures that test traffic traverses the same path as real FCIP traffic. A VLAN tag entry for both the local and remote sides of the route must exist prior to using the **portCmd --traceroute** command. Refer to Managing the VLAN tag table on page 105 for details.

For more information on using traceroute, refer to Using traceroute on page 118.

Bandwidth allocation and restrictions

There are specific bandwidth allocations and restrictions for the FX8-24 blade that are important to review when configuring tunnels and circuits.

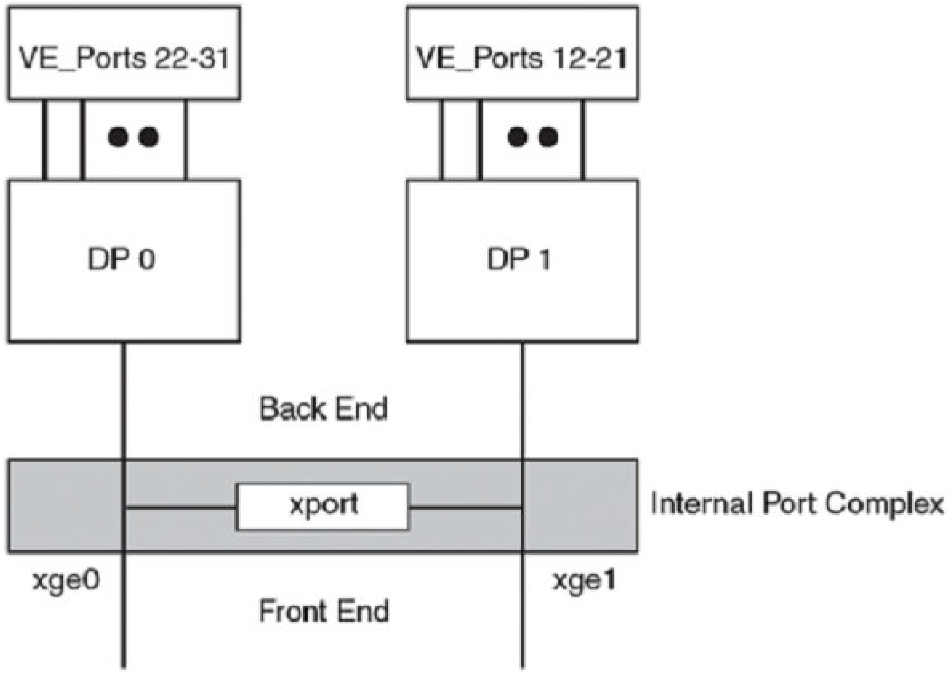
Front-end and back-end bandwidth

The FX8-24 blade contains internal port complex with 1 Gbps ports to support the blade’s VE_Port groups, FCIP Data Processor (DP) complexes, GbE ports, and XGE ports.

Each DP complex, sometimes called an FCIP complex, has 10 Gbps (full-duplex) available bandwidth. Therefore, each VE_Port group (VE_Port 22-31 and VE_Port 12-21) has 10 Gbps of bandwidth available to the internal port complex back-end ports. When the tunnels using VE_Ports in a specific VE_Port group consume the group’s back-end bandwidth, additional circuits cannot be created for those tunnels. The port complex has 10 Gbps of front-end bandwidth available for each of the XGE ports. Tunnels (VE_Ports) cannot consume more than 10 Gbps bandwith over an XGE port. The internal port complex has another 10 Gbps of bandwidth available for the crossport.

The following figure illustrates the internal DP complex with VE_Port groups, internal port complex, front-end and back-end port areas, and the crossport (xport) on an FCX8-24 blade.

FIGURE 7 Internal port and DP complexes on FX8-24 blade



Calculating back-end bandwidth

The following are the ways for configuring the back-end bandwidth for FCIP tunnel and complex:

- To calculate the consumed bandwidth for an FCIP tunnel, round the maximum committed rates for all metric 0 circuits up to the next whole rate (for example 1.5 Gbps becomes 2 Gbps) and add them up. Then add the rounded-up maximum committed rates for all metric 1 circuits. The greater of the two values is the consumed bandwidth for an FCIP tunnel.
- To calculate the total consumed back-end port bandwidth for an FCIP complex, add the consumed bandwidth for each FCIP tunnel in the complex VE_Port group. The total cannot exceed 10 Gbps.
- Back-end bandwidths are always rounded up for each VE_Port group. For example, a circuit defined as 1.5 Gbps will consume 2 Gbps of back-end bandwidth.

Calculating front-end bandwidth

The following are the ways to calculate the front-end bandwidth for a FCIP tunnel and a XGE port:

- To calculate the front-end bandwidth usage on a per-tunnel and per-XGE port basis, add the consumed bandwidth for all metric 0 circuits for a tunnel using xge0 or xge1. Add the total consumed bandwidth for all metric 1 circuits for the tunnel. The greater of the two values is the total front-end port bandwidth usage for an xge0 or xge1 tunnel. Refer to Circuit failover on page 51 for more information on assigning metrics to circuits.
- Each XGE port is allocated 10 Gbps of front-end bandwidth. The total consumed front-end port bandwidth cannot exceed 10 Gbps per XGE port.

Calculating crossport bandwidth

The FCIP DP complexes share only one crossport, so total available bandwidth is 10 Gbps for all VE_Ports on the blade, regardless of the FCIP DP complex to which the VE_Ports belong. For more information on crossports, refer to Crossports on page 33.

- To calculate the consumed crossport bandwidth on a per-tunnel basis, add the consumed bandwidth for all metric 0 circuits in the tunnel that use the crossport. Add the total consumed bandwidth for all metric 1 circuits in the tunnel that use the crossport. The greater of the two values is the total crossport-consumed bandwidth for the tunnel .
- The total crossport-consumed bandwidth is the total of the bandwidth for the tunnels using VE_Ports 12 through 31. The total crossport-consumed bandwidth cannot exceed 10 Gbps.

ARL limits

Bandwidth allocations are subject to the minimum committed rate (-b) and maximum committed rate (-B) set for circuits and tunnels using the Adaptive Rate Limiting (ARL) feature. For more information on ARL and ARL restrictions, refer to Adaptive Rate Limiting on page 16.

Failover circuits and groups

When considering the 10 Gbps bandwidth limit for each FCIP DP complex on an FX8-24 blade, you must also consider failover circuits configured for VE_Ports in each complex. For example, you cannot create a circuit to use an address for a crossport if a 10 Gbps failover circuit is assigned to the VE_Port on that crossport. If the failover circuit were to come online, there would be no available bandwidth for the new circuit.

Failover groups allow you to define a set of metric 0 and metric 1 circuits that are part of a failover group. When all metric 0 circuits in the group fail, metric 1 circuits take over operation, even if there are metric 0 circuits still active in other failover groups. Typically, you would configure only one metric 0 circuit in a failover group. For detailed information, refer to Circuit failover on page 51.

In calculating total bandwidth usage for a tunnel, you must also add the total bandwidth usage per failover group.

To calculate the total bandwidth usage for the failover group in the tunnel, for each failover group (0 through 9), perform the following steps:

1. Add the consumed bandwidth for all metric 0 circuits in the failover group.
2. Add the total consumed bandwidth for all metric 1 circuits in the failover group.

Bandwidth allocation example

The basis for all bandwidth calculations is determining how much bandwidth a given tunnel is consuming. The next step is determining where the tunnel is consuming that bandwidth. You must consider the 10 Gbps limits for back-end ports, front-end ports, and crossports. A tunnel is at least using back-end and front-end port bandwidth. If crossport circuits are configured, then it is using crossport bandwidth as well.

For example, suppose that two 10 Gbps circuits are configured for a tunnel on VE_Port 12. Circuit 0 has a metric of 0 on xge1 and circuit 1 is a failover circuit with a metric of 1 on xge0 (refer to the illustration under Front-end and back-end bandwidth on page 36). Note that configuring circuit 1 on xge0 is a crossport configuration. Although this configuration is allowed, you cannot create additional circuits for VE_Port group 12 through 21 or group 22 through 31 for the following reasons:

- For VE_Port group 12 through 21, VE_Port 12 is consuming the maximum 10 Gbps of allocated back-end port bandwidth. Refer to Calculating Crossport bandwidth on page 38.
- You cannot create a crossport so that VE_Ports 22 through 31 use xge1 because VE_Port 12 is consuming the maximum 10 Gbps of crossport bandwidth for its failover circuit. Refer to Calculating Crossport bandwidth on page 38.
- If VE_Port 12 fails, all 10 Gbps traffic will flow over the crossport and the xge0 front-end port. If additional circuits were already configured for the VE_Port 22 through 31 group, the front-end port bandwidth would exceed the 10 Gbps limit for xge0. Refer to Calculating front-end bandwidth on page 37.

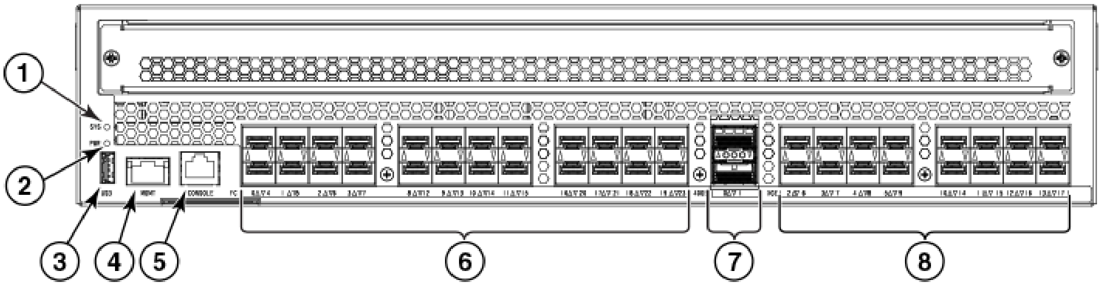
Brocade 7840 Extension Switch

This section provides information on ports, circuits, and tunnels specific to the Brocade 7840 Extension Switch.

NOTE
You cannot connect FCIP tunnels created on a Brocade 7840 switch to interfaces on a Brocade 7500 switch, 7800 switch or FX8-24 blade.

The following figure illustrates the FC ports, 10 GbE, and 40 GbE ports on the Brocade 7840 switch.

FIGURE 8 Brocade 7840 switch ports and status indicators



1. System (SYS) status LED
2. Power (PWR) LED
3. USB port
4. Ethernet management (mgmt) port
5. Serial Console management port
6. FC ports 0-23
7. 40 GbE ports 0-1
8. 1/10 GbE ports 2-17

The Brocade 7840 Extension Switch provides 24 16 Gbps FC ports (FC0-FC23) numbered 0-23 on the switch, two 40 GbE ports (ge0-ge1) numbered 0-1 on the switch, and 16 1/10 GbE ports (ge2-ge17) numbered 2-17 on the switch. Up to 20 VE_Ports are supported for tunnel configurations. Typically, only one VE_Port is needed per remote site.

NOTE
The 40 GbE ports are enabled for configuring IP addresses with the 7840 WAN Rate Upgrade 2 license.

Brocade 7840 DP components and VE_Port distribution

Each Brocade 7840 supports two data processor (DP) complexes. DP complexes are synonymous with FCIP complexes or FCIP engines. Each DP complex contains a data processor (DP) attached to traditional Brocade's switching ASICs, and consists of special purpose hardware for FCIP functions and multicore network processors.

The following figures illustrate components and connections for each DP complex in the Brocade 7840 switch, when the switch is enabled in 10VE or 20VE modes. All 10, 20, and 40 Gbps connections shown in the illustrations are full-duplex and internal in the switch. For more information about 10VE and 20VE port modes, refer to 10VE and 20VE port modes on page 43.

FIGURE 9 DP components and VE_Port distribution in 10VE Mode

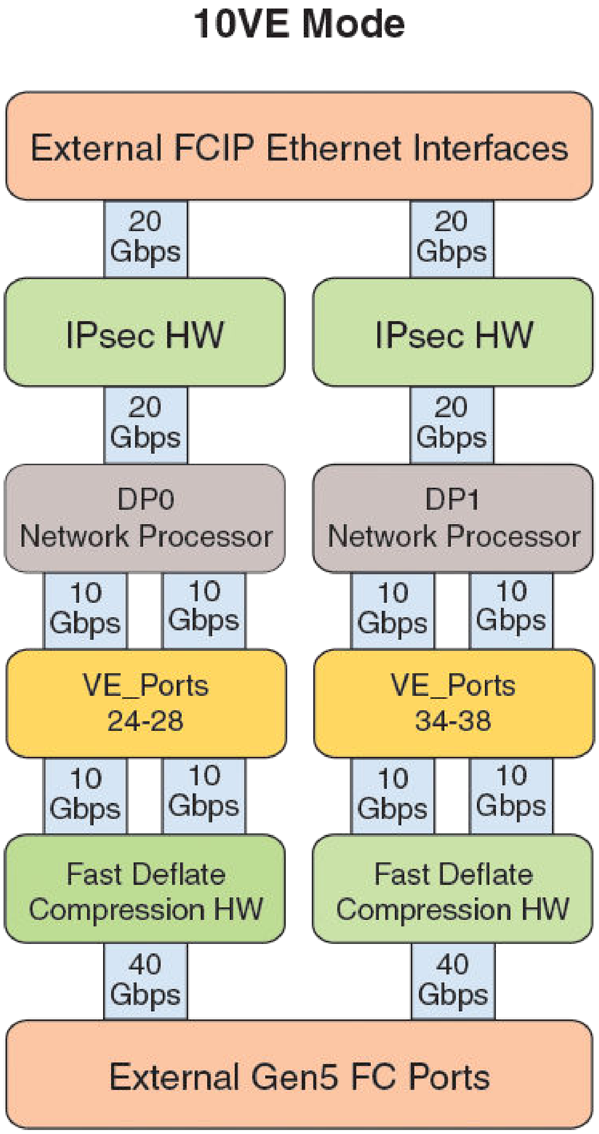
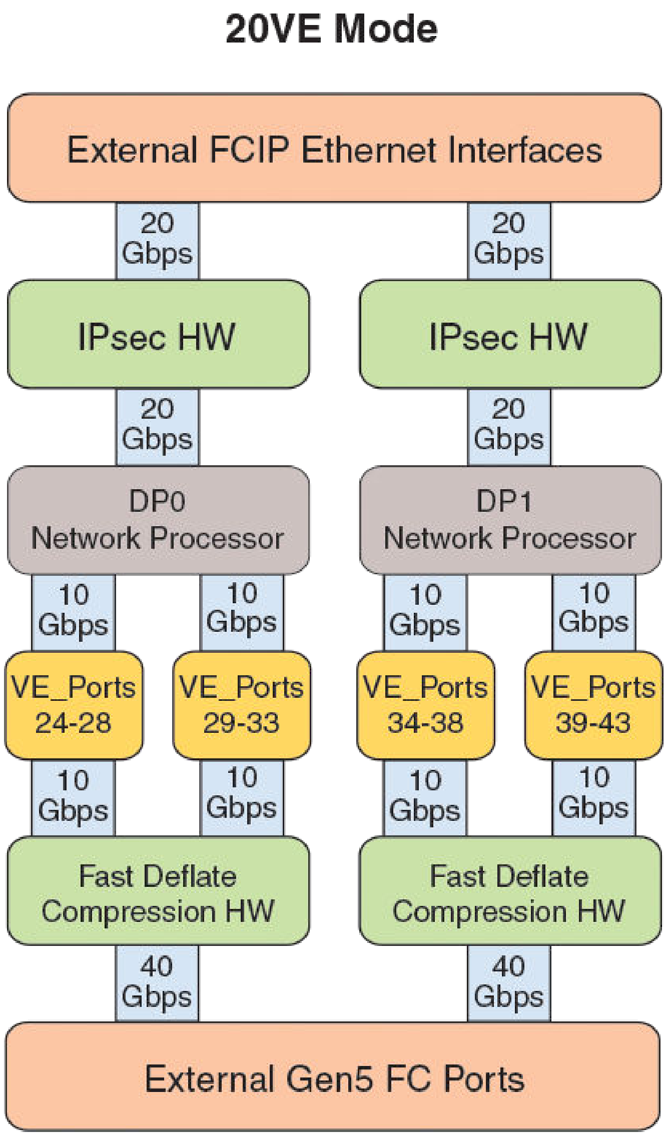


FIGURE 10 DP components and VE_Port distribution in 20VE Mode



As shown in the illustrations:

- There is a 40 Gbps full-duplex connection between the FC switching ASIC and external Gen5 FC ports and each DP.
- Fibre Channel (FC) frames are compressed with the fast deflate compression hardware.
- There are two 10 Gbps full-duplex connections from the fast deflate compression hardware to the VE_Ports and from the VE_Ports to the DP network processor. These 10 Gbps connections can accommodate multiple 10 Gbps or less bandwidth tunnels; however, the maximum bandwidth size of any one tunnel across these internal connections can be no more than 10 Gbps.
- From the network processors, data can be encrypted by the IPsec hardware using high-speed low-latency hardware-based encryptors. Each DP network processor can produce 20 Gbps of FCIP data flow going towards or coming from the external FCIP Ethernet interfaces and the WAN.

If a 4:1 compression ratio is achieved using fast deflate compression, then 80 Gbps is available to exeternal FC ports. The Adaptive Rate Limiting (ARL) aggregate of all circuit maximum values on a single DP complex cannot exceed 40 Gbps. The ARL aggregate of all circuit minimum values for a single DP complex cannot exceed 20 Gbps. All circuits includes all circuits from all tunnels, not just all circuits from a single tunnel.

NOTE
Typical deflate compression may achieve different compression ratios. Brocade makes no promises as to the achievable compression ratios for customer-specific data.

The VE_Port that you use for configuring the FCIP Tunnel also selects the DP complex that will be used for processing. The following lists VE_Port distribution on each DP complex for 10VE and 20VE modes. Refer to 10VE and 20VE port modes on page 43 for more information.

- DP0
  - 10VE and 20VE mode: VE_Ports 24-28
  - 20VE mode only: VE_Ports 29-33
- DP1
  - 10VE and 20VE mode: VE_Ports 34-38
  - 20VE mode only: VE_Ports 39-43

For additional specifications and requirements for 7840 switch ports, tunnels, and circuits, refer to Requirements - 7840 extension switches on page 50.

10 GbE and 40 GbE port considerations

Enhanced 10 GbE and 40 GbE port operation requires special considerations when configuring circuits, tunnels, failover operations, and bandwidth.

For a complete list of tunnel, circuit, and IP address requirements and capacities, refer to Tunnel and circuit requirements on page 47.

Multigigabit circuits

There is no limit on the number FCIP circuits that you can configure on a single 7840 switch port. The maximum committed rate of a single circuit is 10 Gbps, whether configured on a 10 GbE or 40 GbE port.

NOTE
There is no difference in latency or throughput performance for single or multigigabit circuits.

Port grouping

The Brocade 7840 supports eight groups of Ethernet ports. Specific recommendations can be applied to ports within a group to help alleviate traffic congestion problems.

Switch Ethernet ports are numbered from left to right, starting with the 40 GbE ports as 0-1. The 10 GbE ports are numbered 2-17. Refer to the illustration of the switch's port side in Brocade 7840 Extension Switch on page 39 for port numbering. Port numbers contained in port groups are shown in the following table.

TABLE 8 Brocade 7840 switch port groups

Port number	Port group
0, 1, 13, 17	1
2, 6	2
3, 7	3
4, 8	4
5, 9	5
10, 14	6
11, 15	7
12, 16	8

Note that port group 1 contains the two 40 GbE ports (0 and 1) and 10 GbE ports 13 and 17. The remaining port groups contain the 10GbE ports from 2-16. Consider the following when using ports from these port groups:

- A port can block any port in its port group, but It cannot block a port outside of its port group.
- A port could affect another port in the same group due to differences in port speed or if the port is back-pressured due to Ethernet pause from an external switch.

To avoid these effects on ports within the same port group, it is best that you do not mix speeds for ports within the group. Recommendations for the port groups are as follows:

- In port group 1, because the 40 GbE ports are fixed at 40 Gbps, either use the 40 GbE ports or the 10 GbE ports at 10 Gbps or 1 Gbps.
- In port groups 2 through 8, which contain all 10 GbE ports, either configure the ports at 10 Gbps or 1 Gbps.

10VE and 20VE port modes

You can configure the Brocade 7480 switch in either 10VE mode (default) or 20VE mode using the **extncfg** command. This command is disruptive as it requires rebooting the switch.

In 10VE mode, 10 of the 20 VE_Ports are disabled. These are VE_Ports 29-33 and 39-43. Five VE_Ports are enabled on each DP complex as follows:

- DP0 - VE_Ports 24-28
- DP1 - VE_Ports 34-38

In 10VE mode, a VE_Port can use all Fibre Channel bandwidth available to the DP complex where it resides, a maximum of 20 Gbps.

In 20VE mode, all 20 VE_Ports are enabled, 10 on each DP complex as follows:

- DP0 - VE_Ports 24-33
- DP1 - VE_Ports 34-43

In 20VE mode, a single VE_Port on a DP complex can use half the Fibre Channel bandwidth available to the DP complex where it resides, a maximum of 10 Gbps. This option allows use of more VE_Ports, but at a lower maximum bandwidth.

To configure these modes, refer to the *Configuring FCIP* chapter.

Extension Hot Code Load

Extension Hot Code Load (Extension HCL) allows non-disruptive firmware updates on the Brocade 7840 extension switch. This benefits switch operation in mainframe environments by supporting non-stop applications such as data replication and tape backups. Extension HCL supports mainframes by not causing an Interface Control Check (IFCC).

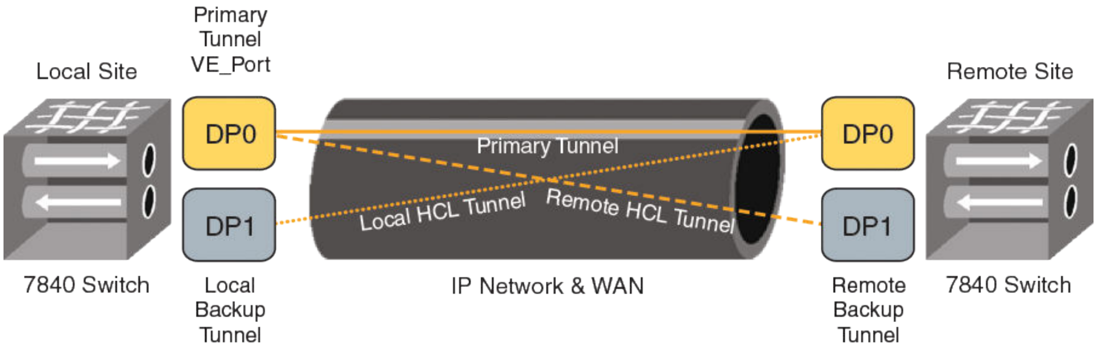
The best practice for using Extension HCL is to update firmware on the switch on one side of the FCIP tunnel, and then update the switch on the other side.

The Brocade 7840 switch has two DP complexes, referred to as DP0 and DP1 (refer to 7840 switch DP components and VE_Port distribution on page 40). An Extension HCL firmware update occurs on one DP complex at a time. When a firmware update is initiated, the process always starts on DP0. As firmware on DP0 is updated, traffic fails over to DP1 to maintain FCIP communication between the local and remote switch.

Extension HCL utilizes three tunnels, as shown in the following illustration, to perform the nondisruptive firmware upload process:

- The primary tunnel (PT) provides FCIP connectivity during normal operations.
- A local backup tunnel (LBT) maintains FCIP connectivity from the remote switch when the local switch DP0 is being upgraded. This tunnel, dormant during non-Extension HCL operations, is created automatically on the local DP1.
- A remote backup tunnel (RBT) maintains FCIP connectivity from the local switch when the remote switch DP0 is being upgraded. This tunnel, dormant during non-Extension HCL operations, is created automatically on the remote DP1.

FIGURE 11 Extension HCL tunnels



The primary tunnel (PT) is what you normally configure to create a tunnel from a VE_Port using the **portcfg fciptunnel** command and appropriate tunnel and circuit parameters. The PT carries traffic through the tunnel to the remote switch. LBT is created automatically on the local DP1 and RBT are created automatically on the remote DP1, as are associated circuits and circuit features. Each end of the LBT and RBT has an associated VE_Port. The local DP0 creates a single VE_Port, which is used for the PT and RBT.

These tunnels are utilized in the following Extension HCL upgrade process:

1. The firmware writes to the backup partition of the control processor.
2. The control processor reboots from the backup partition with the new firmware.
3. The local DP0 is updated with the new firmware using the following process.
    a. The LBT is brought up and DP0 is brought down.
    b. Traffic from the PT is rerouted to DP1 through the LBT so that data traffic can continue between the switches. In-order data delivery is maintained.

   c. All data traffic is flushed from the local DP0 across the IP network and WAN. Buffer queues hold ingress data on DP1 waiting for DP0 to flush.
   d. DP1 queues are unlocked and data transmission resumes. All data remains in order.
   e. DP0 reboots with the new firmware and the configuration is reloaded.
   f. Traffic from the LBT is failed-back to DP0 through the PT .
4. The local DP1 is updated with new firmware using the following process.
   a. The LBT is brought up and DP1 is brought down.
   b. Traffic from the PT is rerouted to DP0 through the LBT so that data traffic can continue between the switches. In-order data delivery is maintained.
   c. All data traffic is flushed from the local DP1 across the IP network and WAN. Buffer queues hold ingress data on DP0 waiting for DP1 to flush.
   d. DP0 queues are unlocked and data transmission resumes. All data remains in order.
   e. DP1 reboots with the new firmware and the configuration is reloaded.
   f. The PT comes back up.
5. After firmware is updated on DP1 and all PTs, LBT, and RBT are offline, the Extension HCL firmware update is complete.

During the update process, tunnels and trunks change state (up or down). The PT provides FCIP connectivity during normal operations. It is up during normal operation and down only during the Extension HCL process. The RBT and LBT is normally up during normal operation, but do not handle traffic. They operate to handle traffic during the Extension HCL process. There may be short periods when both PT and LBT are simultaneously in operation up to ensure in-order delivery of data. RBT handles traffic when the remote switch DP0 undergoes the Extension HCL process. The RBT is visible as a backup tunnel on local DP0.

To configure Extension HCL, refer to Configuring Extension HCL on page 83.

Limitations and considerations

Following are limitations and considerations for using the Extension HCL feature on the Brocade 7840 switch:

- No configuration changes are permitted during the Extension HCL process.
- Extension HCL supports Virtual Fabrics (VF) and FC Routing (FCR with the IR license) and all existing FCIP features.
- Extension HCL was designed for all environments including mainframe FICON XRC and tape and open systems disk replication (EMC SRDF, HDS Universal Replicator, IBM Global Mirror, HP Remote Copy, and others). Extension HCL supports asynchronous and synchronous environments.
- The Brocade 7840 switch has two data processor (DP) complexes: DP0 and DP1. During the HCL process, each DP reloads one at a time, while the other DP remains operational. Consider the following for planning and use of the switch during this process:
  - Because only one DP complex remains operational at a time, the total switch capacity is temporarily diminished by 50 percent.
  - Data is not lost and remains in order. Extension HCL does not cause FICON interface control check (IFCC).
  - The use of Extension HCL requires proper planning. There is large amount of bandwidth available as the Fibre Channel (FC) and FICON side of the switch provides 80 Gbps. In addition, there are typically A and B paths for a total of 160 Gbps in a redundant replication network. This is more than enough bandwidth for most replication networks, even during a path failure or firmware update. Apportioning bandwidth to one DP complex or using only 50 percent of the capacity across both DP complexes reserves adequate bandwidth for high-availability operations. This is considered best practice.
  - The aggregate of all FC and FICON application data passing through the Brocade 7840 cannot exceed 20 Gbps per DP complex multiplied by the achievable compression ratio, or

40 Gbps, whichever is smaller. For example, if 2:1 compression can be achieved, then the storage application could maintain 40 Gbps of throughput across the Extension connection. This is true for both 10VE and 20VE operating modes.

- Although most firmware updates will support Extension HCL, not every Fabric OS release will guarantee firmware capable of using this feature. Refer to the Fabric OS release notes for details.
- The firmware on the switch at each end the FCIP tunnel must be compatible. If not, this will prevent successful tunnel formation when the primary tunnel attempts to come back online or could introduce instability and aberrant behavior.
- Extension HCL does not require any additional communication paths. Although there will be existing FC and FCIP connections used for the normal operation of data replication and tape backup, this is the only requirement.
- Extension HCL is exclusive to the Brocade 7840. It is not compatible with the Brocade 7500 switch, 7800 switch, or the FX8-24 blade.
- Just before the DP complex reset during the upgrade process, an FTRACE capture is triggered in the event that this information is needed post-reset.
- Extension HCL takes advantage of RASlog warnings and error messages (WARN/ERROR).

License options

Important FCIP and FICON extension capabilities of the Brocade 7840 switch require the feature licenses shown in the following table. Use the **licenseshow** command to display license keys and licenses currently installed.

TABLE 9 Brocade 7840 FCIP feature licenses

Feature	Purpose	License (licenseShow output)
WAN Rate Upgrade 1	Increases bandwidth available to all FCIP tunnels configured on the switch from 5 Gbps for the base hardware to 10 Gbps.	LICENSE1
WAN Rate Upgrade 2	Allows unlimited bandwidth for all tunnels configured on the switch. This also enables the 40 GbE ports so that they can be used for configuring IP addresses. **NOTE** You must have an WAN Rate Upgrade 1 license to activate the WAN Rate Upgrade 2 license.	LICENSE2
Advanced FICON acceleration	Enables accelerated tape read/write and IBM z/OS Global Mirror, Teradata, and printer emulation features in FICON environments. Slot-based license.	Advanced FICON Acceleration (FTR_AFA) license
Advanced Extension License	This is enabled on the Brocade 7840 switch at the factory. Required for multiple-circuit tunnels, FCIP Trunking, ARL.	Advanced Extension (FTR_AE) license

For complete information about the licenses described in the preceding table and additional licenses available for the Brocade 7840 switch, refer to the *Fabric OS Software Licensing Guide*.

Path Maximum Transmission Unit discovery

Path Maximum Transmission Unit (PMTU) discovery is supported on Brocade 7840 Extension Switches. On Brocade switches, PMTU is the process of sending Internet Control Message Protocol (ICMP) datagrams of various known sizes across an IP network to determine the supported maximum datagram size.

Based on the largest ICMP Echo Request datagram received, the PMTU discovery process sets the IP MTU for that circuit's IP interface (ipif). Each circuit initiates the PMTU discovery process prior to coming online. This is required because the FCIP circuit may have gone offline due to a link failure, rerouted to a new path, and now has a different MTU. If a circuit bounces, the PMTU discovery process will be initiated when attempting to re-establish the circuit. The PMTU discovery process can result more time for the circuit establishment. The smallest supported MTU size is 1280 bytes. The largest supported IP MTU size on the Brocade 7840 is 9216 bytes; however, PMTU discovery will not discover an MTU greater than 9100 bytes. If the IP network's MTU is known, the best practice is to set it manually in the **portcfg ipif** command. This will avoid values determined by PMTU discovery that are less than the exact MTU of the IP network.

PMTU requires that ICMP is permitted across all IP network devices and the WAN. A rudimentary check would be if you could ping devices across this network. Brocade PMTU discovery uses ICMP Echo Requests. In most cases, only a firewall would block ICMP. If there are no firewalls most likely ICMP is free to traverse the network. If PMTU discovery cannot communicate with the peer switch, the FCIP circuit will not be established.

Enable PMTU discovery by setting the MTU value to "auto" when configuring the ipif for a circuit using the **portcfg ipif** command. Use the **portshow ipif** command to show the configuration of the MTU parameter and **portshow fcipcircuit -d** command to display the actual discovered PMTU value being used. You can also initiate PMTU discovery using the **portcmd -pmtu** command.

Tunnel and circuit requirements

This section describes FCIP tunnel and circuit characteristics, capacities, restrictions, and usage on Brocade extension switches and blades.

General tunnel, circuit, and port requirements

- You can define multiple addresses on Ethernet ports to configure multiple circuits. Multiple circuits can be configured as an FCIP trunk, which provides multiple source and destination addresses to route traffic across an IP network, provide load leveling, and provide failover capabilities.
- The committed rate for a circuit associated with a physical port cannot exceed the rate of the port.
- In a scenario where an FCIP tunnel has multiple circuits of different metrics (0 or 1), circuits with higher metrics (1) are treated as standby circuits and are only used when all lower metric (0) circuits fail. Using Circuit Failover Grouping, you can better control which metric 1 circuits will be activated if a metric 0 circuit fails.
- A circuit defines source and destination IP addresses on either end of an FCIP tunnel.
- If the circuit source and destination IP addresses are not on the same subnet, an IP static route (iproute) must be defined on both sides of the tunnels that designates the gateway IP addresses.
- As a best practice, all tunnel and circuit settings should be identical on both sides of the tunnel. This includes committed bandwidth, IPsec, compression, ARL minimum and maximum, FCIP Fastwrite, OSTP, FICON tunnel, and keepalive timeout values (KATOV).

- VE_Ports or VEX_Ports cannot connect in parallel to the same domain at the same time as Fibre Channel E_Ports or EX_Ports.
- When load-leveling across multiple circuits, the difference between the ARL minimum data rate set on the slowest circuit in the FCIP trunk and the fastest circuit should be no greater than a factor of four. For example, a 100 Mbps circuit and a 400 Mbps circuit will work, but a 10 Mbps and a 400 Mbps circuit will not work. This ensures that the entire bandwidth of the FCIP trunk can be utilized. If you configure circuits with the committed rates that differ by more than a factor of four, the entire bandwidth of the FCIP trunk cannot be fully utilized.

For more information on tunnel and circuit requirements, refer to Limitations on page 18.

Brocade 7800 extension switches

This section describes requirements and specifications for tunnels, circuits, and ports on Brocade 7800 extension switches.

IP addresses and routes:

- You can define up to eight IP addresses for a GbE port.
- You can define up to 32 routes for each GbE port.

VE_Ports, VEX_Ports. and EX_Ports:

- The switch can support eight VE_Ports. VE_Ports are numbered from 16 through 23, therefore up to eight FCIP tunnels can be created. Each FCIP tunnel is identified with a VE_Port number.
- The switch supports EX_Ports and VEX_Ports to avoid the need to merge fabrics.
- VE_Ports do not have to be associated with a particular GbE port.

Bandwidths, maximum and minimum rates:

- The minimum committed rate for a circuit is 10 Mbps.
- The total full-duplex bandwidth limit is 6 Gbps for tunnel connections.
- As a best practice, Fibre Channel traffic through all VE_Port tunnels should not exceed limits set by Adaptive Rate Limiting (ARL). For example, if the link is 500 Mbps, the aggregate of the ARL maximum rates connected to that WAN link can be no more than 500 Mbps. For ingress rates, there is no limit because the FC flow control (BBC) rate limits the incoming data.
- For ARL, configure minimum rates of all the tunnels so that the combined rate does not exceed 6 Gbps for all VE_Ports or the aggregate does not exceed the available WAN bandwidth.
- The maximum trunk capacity is 6 Gbps.

Circuits:

- The switch contains up to six GbE ports. You can configure up to six circuits per tunnel (VE_Port) spread out over any of these ports.
- A limit of four FCIP circuits can be configured on a single GbE port. Each circuit requires a unique IP address.
- The total circuits per switch cannot exceed 24 (a total of four circuits for all GbE ports).
- A single FCIP circuit cannot exceed 1 Gbps capacity.

Although a Brocade 7800 switch only contains six GbE ports, eight VE_Ports (16-23) are available for tunnels. This is because in most cases one VE_Port is used per site and there may be multiple sites connected to the switch. As another case, when using logical fabrics, a VE_Port can be used per logical switch. The VE_Ports in different logical switches can share a single GbE port located on the default switch. Refer to the example distribution in the following table for how the Brocade 7800 VE_Ports GbE ports might be used for port sharing in circuit configurations. Note that no more than four VE_Ports (tunnels) are using the same GbE port, which is the standard limit for this switch.

TABLE 10 Example VE_Ports versus GbE ports used on the Brocade 7800 switch

VE_Port	GbE Ports
16	GE0, GE1, GE2, GE3
17	GE0, GE1, GE2, GE3
18	GE0, GE1, GE2, GE3
19	GE0, GE1, GE2, GE3
20	GE4, GE5
21	GE4, GE5
22	GE4, GE5
23	GE4, GE5

Refer to Ethernet Port sharing on page 95 for more information on port sharing in a Virtual Fabrics environment.

FX8-24 extension blades

This section lists requirements and specifications for tunnels, circuits, and ports on FX8-24 extension blades.

IP addresses and routes:

- You can define up to 10 IP addresses for a 10 GbE port and an additional 10 addresses on crossports when operating in 10 Gbps mode.
- You can define up to eight IP addresses (0 through 8 minus the default IPv6 “link-local” address) for a 1 GbE port.

VE_Ports, VE_Port groups, VEX_Ports:

- An FX8-24 blade can support 20 VE_Ports, and therefore 20 FCIP tunnels.
- There are two VE_Port groups. DP1 controls ports numbered 12 through 21 and DP0 controls ports numbered and 22 through 31.
- Each FCIP tunnel is identified with a VE_Port number.
- VE_Ports do not have to be associated with a particular Ethernet port.
- The blade also supports VEX_Ports to avoid the need to merge fabrics.
- VE_Port versus Ethernet port usage depends on the blade operating mode as follows:
  - 1 Gbps mode: VE_Ports 12 through 21 are available to use GbE ports 0 through 9. VE_Ports 22-31, xge0, and xge1 are not available.
  - In 10 Gbps mode, VE_Ports 12 through 21 are available to use xge1; VE_Ports 22 through 31 are available to use xge0. GbE ports 0 through 9 are not available.
  - In 10 Gbps mode, you can also configure VE_Ports 12 through 21 to use port xge0 as a crossport and VE_Ports 22 through 31 to use port xge1 as a crossport.
  - In dual mode, VE_Ports 12 through 21 are available to use GbE ports 0 through 9; VE_Ports 22 through 31 are available to use xge0. Port xge1 is not available.

Circuits:

- A limit of 20 FCIP circuits can be configured per VE_Port group (12 through 21 or 22 through 31) when using a 10 GbE port. For the 20 circuits, 10 are configured on local ports and 10 on crossports
- You can configure up to 10 circuits for an FCIP trunk (VE_Port).
- The FX8-24 blade contains two 10 GbE ports. You can define up to 10 circuits per FCIP trunk spread across the 10 GbE ports.
- A limit of 10 FCIP circuits can be configured on a single 10 GbE port. Each circuit requires a unique IP address.
- The blade contains ten 1 GbE ports. You can define up to 10 circuits per FCIP trunk spread across the GbE ports.
- A limit of four FCIP circuits can be configured on a single 1 GbE port. Each circuit requires a unique IP address.

Bandwidths, maximum and minimum rates:

- For an FX8-24 blade with a VE_Port group on a 10 GbE port, the sum of the maximum committed rates of that group's circuits cannot exceed 10 Gbps.
- For ARL, configure minimum rates of all the tunnels so that the combined rate does not exceed 20 Gbps for all VE_Ports on the blade.
- For ARL, you can configure maximum rate of 10 Gbps for all tunnels over a single 10 GbE port and 10 Gbps for any single circuit.
- The minimum committed rate for a circuit is 10 Mbps.
- A circuit between 1 GbE ports cannot exceed the 1 Gbps capacity of the interfaces rate.

For additional considerations on multigigabit circuits configured on 10 GbE ports, refer to Multigigabit circuits FX8-24 blades on page 33.

Brocade 7840 extension switches

This section lists requirements and specifications for tunnels, circuits, and ports on Brocade 7840 extension switches.

IP addresses and routes:

- You can configure maximum 60 IP addresses per DP complex.
- You can define up to 128 routes per GbE port; however, you can only define 120 routes per DP. The DP limit would take precedence.

VE_Ports, VE_Port groups, and VEX_Ports:

- There are two VE_Port groups in 10VE mode. DP0 controls VE_Ports 24-33 and DP1 controls VE_Ports 34-43. Each port group can share 20 Gbps.
- You can have a maximum 20 VE_Ports on the switch. In VE10 mode (default), only 10 VE_Ports are enabled. In VE20 mode, all 20 VE_Ports are enabled. The default is 10VE mode.
- There are four VE_Port groups in 20VE mode. DP0 controls VE_Ports 24-28 and VE_Ports 29-33. DP1 controls VE_Ports 34-38 and VE_Ports 39-43. Each port group can share 10 Gbps.
- Each DP complex supports 10 VE_Ports in 20VE mode and therefore 10 distinct FCIP tunnels.
- Each DP complex supports 5 VE_Ports in 10VE mode and therefore 5 distinct FCIP tunnels.
- VE_Ports do not have to be associated with a particular Ethernet port.
- VE_Ports cannot connect in parallel to the same domain at the same time as Fibre Channel E_Ports or EX_Ports.
- VEX_Ports are not supported on this platform.

Bandwidths, maximum and minimum rates:

- For a VE_Port group, the sum of the minimum committed rates of that group's circuits cannot exceed 10 Gbps when the VE_Ports are in 20VE mode and 20 Gbps when ports in 10VE mode.
- The minimum committed rate for all VE_Ports in one DP complex cannot exceed 20 Gbps. The maximum rate for all VE_Ports in one DP complex cannot exceed 40 Gbps.
- The minimum committed rate for a circuit is 20 Mbps.
- The maximum committed rate for a circuit is 10 Gbps.
- With compression, total bandwidth cannot exceed 80 Gbps (40 Gbps per DP) on the Fibre Channel side.
- The difference between the guaranteed (minimum) and maximum bandwidth for a tunnel cannot exceed the 5:1ratio.

Circuits:

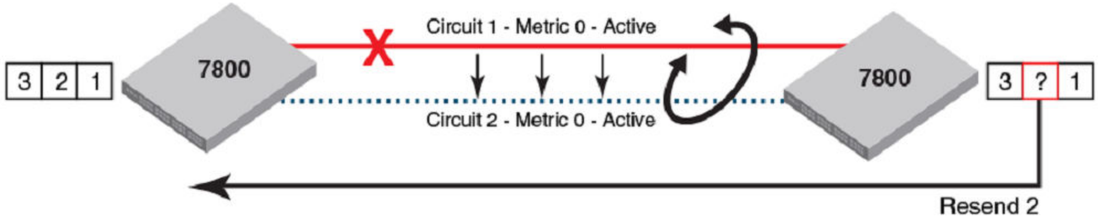
- There is no limit for the number of circuits that you can configure on a Ethernet port. Each circuit requires a unique IP address pair.
- In 20VE mode, 10 VE_Ports are controlled by each DP complex. With a maximum of four circuits per VE_Port, you can configure up to 40 circuits per VE_Port group.
- In 10VE mode (default), five VE_Ports are controlled by each DP complex. With a maximum of four circuits per VE_Port, you can configure up to 20 circuits per VE_Port group.
- You can configure a maximum of four circuits for an FCIP trunk (VE_Port).

For additional considerations on multigigabit circuits configured on Ethernet ports, refer to Multigigabit circuits 7840 switches on page 42.

Circuit failover

Each FCIP circuit is assigned a metric, either 0 or 1, which is used in managing failover from one circuit to another. FCIP Trunking with metrics uses lossless link loss (LLL), and no in-flight data is lost during the failover. If a circuit fails, FCIP Trunking first tries to retransmit any pending send traffic over another lowest metric circuit. In the following figure, circuit 1 and circuit 2 are both lowest metric circuits. Circuit 1 has failed, and transmission fails over to circuit 2, which has the same metric. Traffic that was pending at the time of failure is retransmitted over circuit 2. In-order delivery is ensured by the receiving extension switch or blade.

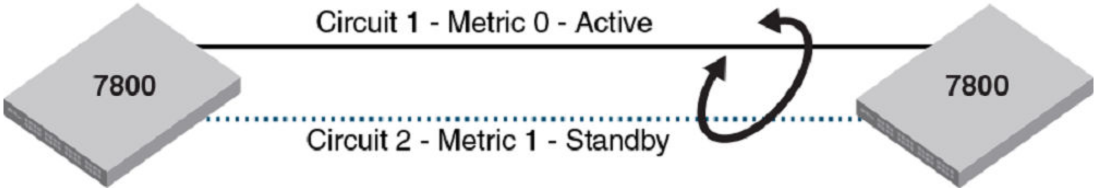
FIGURE 12 Link loss and retransmission over peer lowest metric circuit



NOTE
Modifying a circuit metric disrupts traffic.

In the following figure, circuit 1 is assigned a metric of 0, and circuit 2 is assigned a metric of 1. Both circuits are in the same FCIP tunnel. In this case, circuit 2 is not used until no lowest metric circuits are available. If all lowest metric circuits fail, then the pending send traffic is retransmitted over any available circuits with the higher metric. Failover between like metric circuits or between different metric circuits is lossless.

FIGURE 13 Failover to a higher metric standby circuit



Only when all metric 0 circuits fail do available metric 1 circuits cover data transfer. If the metric 1 circuits are not identical in configuration to the metric 0 circuits, then the metric 1 circuits will exhibit a different behavior. Additionally, if the metric 1 WAN path has different characteristics, these characteristics define the FCIP behavior across the metric 1 circuits. Consider configuring circuit failover groups to avoid this problem.

Circuit Failover Grouping

With circuit failover groups, you can better control which metric 1 circuits will be activated if a metric 0 circuit fails. To create circuit failover groups, you define a set of metric 0 and metric 1 circuits that are part of the same failover group. When all metric 0 circuits in the group fail, metric 1 circuits will take over data transfer, even if there are metric 0 circuits still active in other failover groups.

Typically, you would only define one metric 0 circuit in the group so that a specific metric 1 circuit will take over data transfer when the metric 0 circuit fails. This configuration prevents the problem of the tunnel operating in a degraded mode, with fewer than the defined circuits, before multiple metric 0 circuits fail.

Considerations and limitations

Circuit failover groups operate under the following conditions:

- Each failover group is independent and operates autonomously.
- All metric 0 circuits in a group must fail before the metric 1 circuits are used.
- All metric 1 circuits in a group are used if all metric 0 circuits in the group fail or there is no metric 0 circuit in the group.
- Circuits can be part of only one failover group
- Circuit failover groups are only supported by Fabric OS v7.2.0 or later.
- Both ends of the FCIP tunnel must have the same circuit failover groups defined.
- When a tunnel activates or circuits are modified, tunnel and circuit states will indicate a misconfiguration error if circuit failover group configurations are not valid.
- Modifying of the failover group ID is a disruptive operation, similar to modifying the metric.
- Circuit failover groups are not used to define load balancing over metric 0 circuits (*only* failover rules). Circuits of metric 0 will be load balanced over metric 1 circuits regardless of failover grouping.
- When no FCIP circuit failover groups are defined, failover reverts to the default operation: all metric 0 circuits must fail before failing over to metric 1 circuits. To change the default failover operation, a failover group should include at least one metric 0 and at least metric 1 circuit.
- A valid failover group requires at least one metric 0 circuit and at least one metric 1 circuit; otherwise, a warning displays. If there is no metric 0 circuit and only a metric 1 circuit, the metric 1 circuit will be used, regardless of whether there are metric 0 circuits in another failover group.
- The number of valid failover groups defined per tunnel is limited by the number of circuits that you can create for the switch model as follows:

- For an FX8-24 blade, you can configure up to 5 valid groups on a 10-circuit tunnel.
- For a Brocade 7800 switch, you can have up to 3 valid groups because you can configure 6 circuits per tunnel.
- For a Brocade 7840 switch, you can have up to 2 valid groups because you can configure 4 circuits per tunnel.

- Consider available WAN bandwidth requirements when configuring failover circuit groups. Refer to Bandwidth calculation during failover on page 58.

Examples of circuit failover in groups

The following table illustrates circuit failover in a tunnel with two failover groups, each with two circuits. All data through the tunnel is initially load balanced over circuits 1 and 2. The following occurs during circuit failover:

- If circuit 1 fails, circuit 3 becomes active and data is load balanced over circuits 2 and 3.
- If circuit 2 fails, circuit 4 becomes active and data is load balanced over circuits 1 and 4.
- If both circuit 1 and 2 fail, circuit 3 and 4 become active and data is load balanced over both circuits.

TABLE 11 Tunnel with two failover groups with two circuits

Circuits in tunnel	Failover group ID	Circuit bandwidth	FSPF link cost if circuit goes offline	In use for tunnel data
Circuit 1 Metric 0	1	500 Mb	1,500	If active, yes.
Circuit 2 Metric 0	2	1000 Mb	1,000	If active, yes.
Circuit 3 Metric 1	1	500 Mb	1,500	Only when circuit 1 fails.
Circuit 4 Metric 1	2	1000 Mb	1,000	Only when circuit 2 fails.

The following table illustrates circuit failover in a tunnel with one failover group containing three circuits. In this case, failover occurs as if circuits are not part of a failover group. Circuits 2 and 3, both with metric 1, become active only after circuit 1 with metric 0 fails.

TABLE 12 Tunnel with one failover group with three circuits

Circuits in tunnel	Failover group ID	Circuit bandwidth	FSPF link cost if circuit goes offline	In use for tunnel data
Circuit 1 Metric 0	1	1000 Mb	1,000	If active, yes.
Circuit 2 Metric 1	1	500 Mb	1,500	Only when circuit 1 fails.
Circuit 3 Metric 1	1	500 Mb	1,500	Only when circuit 1 fails.

The following table illustrates circuit failover in a tunnel with circuits in failover groups and circuits that are not part of failover groups. In this configuration, all data is initially load balanced over circuit 1, circuit 2, and circuit 3 (when they are all active). The following occurs during circuit failover:

- If circuit 1 fails, circuit 4 becomes active and data is load balanced over circuit 2, circuit 3, and circuit 4.

Reason: Circuit 1 fails over to circuit 4 (both are in failover group 1) and circuit 3 is active with 500 Mb bandwidth.

- If circuit 2 fails, data is load balanced over circuit 1 and circuit 3, and no other circuit becomes active.

  Reason: Circuits 1 and 3 are the only active circuits because circuits 4 and 5 only become active when circuits 1 or 3 fail.
- If circuit 2 and circuit 3 fail, circuit 5 becomes active and data is load balanced over circuit 1 and circuit 5.

  Reason: Ungrouped circuits 2 and 3 fail over to ungrouped circuit 5, which has a metric of 0.
- If circuit 1, circuit 2, and circuit 3 fail, circuit 4 and circuit 5 become active and data is load balanced over both.

  Reason: Circuit 1 fails over to circuit 4, which is the failover circuit for group 1 with a metric of 0. Ungrouped circuit 5 is the failover circuit for ungrouped, failed circuits 2 and 3.

TABLE 13 Tunnel with failover groups and non-grouped circuits

Circuits in tunnel	Failover group ID	Circuit bandwidth	FSPF link cost if circuit goes offline	In use for tunnel data
Circuit 1 Metric 0	1	500 Mb	1,500	If active, yes.
Circuit 2 Metric 0	Not defined.	500 Mb	1,500	If active, yes.
Circuit 3 Metric 0	Not defined.	500 Mb	1,500	If active, yes.
Circuit 4 Metric 1	1	500 Mb	1,500	Only when circuit 1 fails.
Circuit 5 Metric 1	Not defined.	1000 Mb	1,000	Only when circuits 2 and 3 fails.

Configuring circuit failover groups

Configure failover groups by specifying a metric with the -x|--metric# option and the failover group with the -g|--failover-group ID option, as in the following commands:

- **portcfg fciptunnel slot/ve_port create** --remote-ip *destination address* --local-ip *source address* -x|--metric *[0/1]* -g|--failover-group *[0-9]*
- **portcfg fcipcircuit slot/ve_port create** *cir#* --remote-ip *destination address* --local-ip *source address* -x|--metric *[0/1]* -g|--failover-group *[0-9]*
- **portcfg fcipcircuit slot/ve_port modify** *cir#* --remote-ip *destination address* --local-ip *source address* -x|--metric *[0/1]* -g|--failover-group *[0-9]*

Pay attention to the following considerations for configuring the failover group ID:

- Modifying the failover group ID is a disruptive operation, similar to modifying the metric.
- The ID is used to designate backup (metric 1) circuits to be activated if a metric 0 circuit fails.
- The ID must be an integer value from 0 through 9. ID 0 is used to designate the default failover group (or no failover group).
- The IDs for a failover group must match for the circuit at both ends of the tunnel.

Configuration examples

The following example shows the configuration of two failover groups for VE_Port 22 containing two circuits each. Note that circuit 0 is typically created automatically when the tunnel is created.

```
portcfg fcipcircuit 8/22 create 0 --remote-ip 1.42.128.93 --local-ip 1.42.128.23 -x 0 -g 0 -b 5000000
-B 5000000
portcfg fcipcircuit 8/22 create 1 --remote-ip 1.42.128.94 --local-ip 1.42.128.24 -x 0 -g 1 -b 2750000
-B 2750000
portcfg fcipcircuit 8/22 create 2 --remote-ip 1.42.128.95 --local-ip 1.42.128.25 -x 1 -g 0 -b 4000000
-B 4000000
portcfg fcipcircuit 8/22 create 3 --remote-ip 1.42.128.96 --local-ip 1.42.128.26 -x 1 -g 1 -b 5000000
-B 5000000
```

Entering the **portshow fciptunnel -c** command for the configuration, displays the following output.

```
-------------------------------------------------------------------------------
Tunnel Circuit  OpStatus  Flags    Uptime  TxMBps  RxMBps ConnCnt CommRt Met/G
-------------------------------------------------------------------------------
8/22   -         Up      cft----   26m51s    0.00    0.00    1      -      -/-
8/22   0 8/xge0  Up      ---4--s   26m51s    0.00    0.00    1  5000/5000  0/-
8/22   1 8/xge0  Up      ---4--s   26m51s    0.00    0.00    1  2750/2750  0/1
8/22   2 8/xge0  Up      ---4--s     2m7s    0.00    0.00    1  4000/4000  1/-
8/22   3 8/xge0  Up      ---4--s       0s    0.00    0.00    1  5000/5000  1/1
-------------------------------------------------------------------------------
```

Note in the output that "-" displays for the group (G) to indicate the default failover group 0 or that no failover group is configured.

If you do not configure at least one metric 0 and one metric 1 circuit for a failover group, the Opstatus column of the output will display a failover group warning (FGrpWrn) as in the following output. The warning occurred because only a circuit with metric 0 was created for failover group 0 and group 1. Note that FGrpWrn will not be issued for the default group unless a failover group is configured but not complete with a metric 0 and metric 1 circuit.

NOTE
For the 7840 switch, 'UpWrn' displays if a tunnel is not specified in the **portshow fciptunnel** command, such as in `portshow fciptunnel all -c -h`, while 'Online Warning' displays is if a tunnel is specified, such as in `portshow fciptunnel 24`.

```
-------------------------------------------------------------------------------
Tunnel Circuit  OpStatus  Flags    Uptime  TxMBps  RxMBps ConnCnt CommRt Met/G
-------------------------------------------------------------------------------
8/22   -         FGrpWrn cft----   20m26s    0.00    0.00    1      -      -/-
8/22   0 8/xge0  Up      ---4--s   20m26s    0.00    0.00    1  5000/5000  0/-
8/22   1 8/xge0  Up      ---4--s       3s    0.00    0.00    2  2750/2750  0/1
-------------------------------------------------------------------------------E
```

Entering the **portshow fciptunnel** command for a specific tunnel, such as **portshow fciptunnel 8/22 -c** for tunnel 22, displays detailed information for each circuit in the tunnel. Entering the **portshow fcipcircuit** command, such as **portshow fcipcircuit 8/22 1** for circuit 1, displays detailed information for the specific circuit. This information includes the failover group ID configured for the circuits. If (Not Config/Active) displays for Failover Group ID, the default group ID of 0 was used.

For more information on the **portcfg fcipcircuit**, **portcfg fciptunnel**, and **portshow** commands, refer to the *Fabric OS Command Reference*.

10 GbE Lossless Link Loss (FX8-24 blade)

Circuit failover is supported between 10 GbE circuits on FX8-24 blades when both 10 GbE ports are on the same logical switch and are operating in 10 Gbps mode. You can configure higher metric circuits for failover from lower metric circuits (refer to Circuit failover on page 51). You can also configure IP addresses for a failover crossport. Crossports are IP addresses (and routes) that belong to the other 10 GbE port's VE group. The crossport for xge0 is xge1 and the crossport for xge1 is xge0. For more information on crossports and configuring crossports, refer to Crossports on page 33.

LLL is supported per VE_Port on the VE_Port's DP complex. Because a VE_Port cannot span GbE and 10 GbE interfaces, neither can LLL. LLL is supported on both GbE and 10 GbE interfaces, just not together.

Benefits and limitations of 10 GbE lossless link loss (LLL) failover include the following:

- LLL provides failover to protect against link or network failure and 10 GbE port disable.
- Data will not be lost due to failover.
- Failover supports active-passive and active-active configurations.
- Dual mode is not supported for 10 GbE port failover.
- Failover does not protect against failure of an FCIP DP complex.
- Disabling a VE_Port will not use LLL. In this case, route failover will occur at the FC level based on APT policy, if there is another route available, and may cause loss of FC frames.

NOTE
All circuits and data must belong to a single VE_Port to benefit from LLL.

Configuring failover

There are two types of configuration supported:

- Active-active - Data will be sent on both 10 GbE ports to initiate weighted balancing of the batches across the FCIP trunk's circuits.
- Active-passive - Data fails over using LLL to a passive circuit (one with a higher metric) if all active lower metric circuit paths fail.

You must establish a metric for failover circuits. If no metric is provided, circuit data will be sent through both ports and the load will be balanced. Circuits have a default metric of 0. A metric of 1 is required for a standby (passive) circuit.

Active-active configuration

The following example shows an active-active configuration in which two circuits are configured with the same metric, one circuit going over xge0 and the other circuit going over the crossport using xge1 as the external port. The metric values of both the circuits are the same (default value), so both circuits send data. The load is balanced across these circuits. The effective bandwidth of the tunnel in this example is 2 Gbps.

1. Configure an IP address on interface xge0.

```
portcfg ipif 8/xge0 create 192.168.11.20 netmask 255.255.255.0 mtu 1500
```

2. Configure an IP address on crossport interface xge1.

```
portcfg ipif 8/xge1 create 192.168.10.10 netmask 255.255.255.0 mtu 1500 -x
```

3. Create a tunnel with one circuit going over xge0.

```
portcfg fciptunnel 8/22 create --remote-ip 192.168.11.20 --local-ip 192.168.11.21
-b 2750000 -B 2750000
```

4. Add another circuit, going over crossport xge1, to the tunnel.

   ```
   portcfg fcipcircuit 8/22 create 1 --remote-ip 192.168.10.10 --local-ip
   192.168.10.11 1000000
   ```

5. Display local and crossport interface details for xge0.

   ```
   portshow ipif 8/xge0
   ```

NOTE
If the source and destination addresses are on different subnets, you must configure IP routes to the destination addresses. Refer to Configuring an IP route on page 66.

Active-passive configuration

The following example shows an active-passive configuration in which two circuits are configured with different metrics, one circuit going over xge0 and the other circuit going over the crossport using xge1 as the external port. In this example, circuit 1 is a failover circuit because it has a higher metric. When circuit 0 goes down, the traffic is failed over to circuit 1. The effective bandwidth of the tunnel in this example is 1 Gbps.

1. Configure an IP address on interface xge0.

   ```
   portcfg ipif 8/xge0 create 192.168.11.20 netmask 255.255.255.0 mtu 1500
   ```

2. Configure an IP address on crossport interface xge1.

   ```
   portcfg ipif 8/xge1 create 192.168.10.10 netmask 255.255.255.0 mtu 1500 -x
   ```

3. Create a tunnel with one circuit going over xge0.

   ```
   portcfg fciptunnel 8/22 create --remote-ip 192.168.11.21 --local-ip 192.168.11.20 -
   b 2750000 -B 2750000 --metric 0
   ```

4. Add another circuit, going over crossport xge1, to the tunnel.

   ```
   portcfg fcipcircuit 8/22 create 1 --remote-ip 192.168.10.10 --local-ip
   192.168.10.11 1000000 --metric 1
   ```

5. Display local and crossport interface details for xge0.

   ```
   portshow ipif 8/xge0
   ```

NOTE
If the source and destination addresses are on different subnets, you must configure IP routes to the destination addresses. Refer to Configuring an IP route on page 66.

Failover in TI zones

In Traffic Isolation (TI) zone configurations with failover enabled, non-TI zone traffic will use the dedicated path if no other E_Port or VE_Port paths exist through the fabric or if the non-dedicated paths are not the shortest paths. Note that a higher-bandwith tunnel with multiple circuits will become the shortest path compared to a tunnel with one circuit. A TI zone cannot subvert the Fabric Shortest Path First (FSPF) protocol. Data will never take a higher cost path because a TI zone has been configured to do so. It may be necessary to configure explicit link cost to produce Equal-Cost Multi-Path (ECMP) or to prevent FCIP trunk costs from changing in the event that a circuit goes offline.

Bandwidth calculation during failover

The bandwidth of higher metric circuits is not calculated as available bandwidth on an FCIP tunnel until all lowest metric circuits have failed.

Assume the following configurations for circuits 0 through 3:

- Circuits 0 and 1 are created with a metric of 0. Circuit 0 is created with a maximum transmission rate of 1 Gbps, and circuit 1 is created with a maximum transmission rate of 500 Mbps. Together, circuits 0 and 1 provide an available bandwidth of 1.5 Gbps.
- Circuits 2 and 3 are created with a metric of 1. Both are created with a maximum transmission rate of 1 Gbps, for a total of 2 Gbps. This bandwidth is held in reserve.

The following actions occur during circuit failures:

- If either circuit 0 or circuit 1 fails, traffic flows over the remaining circuit while the failed circuit is being recovered. The available bandwidth is still considered to be 1.5 Gbps.
- If both circuit 0 and circuit 1 fail, there is a failover to circuits 2 and 3, and the available bandwidth is updated as 2 Gbps.
- If a low metric circuit becomes available again, the high metric circuits return to standby status, and the available bandwidth is updated again as each circuit comes online. For example, if circuit 0 is recovered, the available bandwidth is updated as 1 Gbps. If circuit 1 is also recovered, the available bandwidth is updated as 1.5 Gbps.

Configuring FCIP

Configuration preparation

Before you begin to configure FCIP, do the following:

- Determine the amount of bandwidth that will be required for the remote data replication (RDR), FICON, or tape application to be deployed.
- Confirm that the WAN link has been provisioned and tested for integrity.
- Make sure that cabling within the data center has been completed.
- Make sure that switches and other devices have been physically installed and powered on.
- Make sure you have admin access to all switches and blades you need to configure.
- For the Brocade 7800 switch, determine if copper or optical ports will be used for GbE ports 0 and 1.

- For the FX8-24 blade, determine which of the three possible GbE or XGE port operating modes will be used.
- Determine which 10 GbE crossports on FX8-24 blades should get active-active or active-passive configurations.
- For the Brocade 7840 switch, determine the VE_Port operating modes that will be used (10VE mode or 20VE mode).
- Determine which Ethernet ports will be used. The Ethernet ports on the Brocade 7840 switch are in groups and connections should be spread across the groups and not within the groups if possible.
- Obtain subnets and assign IP addresses for each circuit endpoint that you intend to use, plus the netmask and IP MTU size. The IP MTU size may be smaller than 1500 if there is an IPsec device or similar device in the FCIP path. If the IP MTU is larger than 1500, use the following guidelines for your extension product:
  - For the Brocade 7800 switch and FX8-24 blade, use 1500.
  - For the Brocade 7840 device, the IP MTU size must be at least 1280. If the supported maximum IP MTU size in the network is larger than 9216, the IP MTU of the Brocade 7840 should be 9216. For the Brocade 7840, you can use Path MTU Discovery to automatically set the IP MTU size for the circuit's IP interface. Refer to Path Maximum Transmission Unit discovery on page 47 for more information.
- Determine the gateway IP address as needed for each route across the WAN. The gateway IP address will be on the same IP subnet as the subnet used for the FCIP IPIF interface that will use that gateway. The route will be the subnet and netmask on the remote side.
- Determine if there is any reason to turn off selective acknowledgement (SACK). Because SACK improves performance for most installations, it is turned on by default.
- Determine the VE_Port numbers you want to use. The VE_Port numbers serve as tunnel IDs. Typically, the first one is used.
- Determine source and destination IP addresses for circuit 0, and the minimum and maximum rates for ARL. These values are set by the **portCfg fciptunnel create** command. If ARL is not being used, then only the committed rate is required for circuit 0.
- Determine how many additional FCIP circuits you want to create. You will need the source and destination IP addresses for each circuit, and the minimum and maximum rates for ARL, or the committed rate if not using ARL. You will need to know if you intend to assign metrics to circuits so that lower metric circuits fail over to circuits with higher metrics. For all circuits except circuit 0, these values are set by the **portCfg fcipcircuit create** command.
- When configuring tunnels to support large numbers of devices, consider memory limitations of the extension switch or blade if you are enabling any type of emulation feature, such as FCP or FICON. If too many devices are present or activated at one time, acceleration operations can be negatively impacted. Refer to Memory use limitations for large-device tunnel configurations on page 23.

Configuration steps

Use the following major steps for configuring FCIP on extension switches and blades:

1. Persistently disable VE_Ports.
2. If required, configure VEX_Ports.
3. Set the media type or operating mode:
   - For the Brocade 7800 switch, set the media type for GbE ports 0 and 1.
   - For the FX8-24 blade, set the GbE and XGE port operating mode.
   - For the Brocade 7840 switch, configure 10VE or 20VE operating mode.
4. Create an IP interface (IPIF) for each circuit that you want on a port by assigning an IP address, netmask, and an IP MTU size to an Ethernet port using the **portCfg ipif** command. Refer to Configuring an IPIF on page 65.

5. Create one or more IP routes to a port if required using the **portCfg iproute** command. Refer to Configuring an IP route on page 66.
6. Test the IP connection using the **portCmd --ping** command.

NOTE
When using VLANS, VLAN tagging ensures that test traffic traverses the same path as real FCIP traffic. A VLAN tag entry for both the local and remote sides of the route must exist prior to using the **portCmd --ping** command. Refer to Managing the VLAN tag table on page 105 for details.

7. Create FCIP tunnels using the **portCfg fciptunnel** command. Refer to Creating an FCIP tunnel on page 68.
8. Create FCIP circuits (after circuit 0) and enable or disable features using the **portCfg fcipcircuit** command. Refer to Creating additional FCIP circuits on page 82.

NOTE
Configuring a tunnel automatically configures circuit 0 for the tunnel, although you can use **portcfg fciptunnel** parameters to create a blank tunnel.

9. Persistently enable the VE_Ports.

Setting VE_Ports to persistently disabled state

It is strongly recommended to persistently disable VE_Ports while tunnel configuration is in progress. This will prevent unwanted fabric merges from occurring until the FCIP tunnel is fully configured. You must change the state of the VE_Ports from persistently enabled to persistently disabled. Once the FCIP tunnels have been fully configured on both ends of the tunnel, you can persistently enable the ports.

1. Enter the **portCfgShow** command to view ports that are persistently disabled.
2. Enter the **portCfgPersistentDisable** command to disable any VE_Ports that you will use in the FCIP tunnel configuration.

Disabling ports when FMS Mode is enabled

If you enter **portCfgPersistentDisable** and receive "command not allowed in fmsmode" or "command not found" messages, FMS mode may be enabled. You cannot use the **portCfgPersistentDisable** or **portCfgPersistentEnable** commands with FMS mode enabled. Use the **portDisable** and **portEnable** commands instead.

You can determine if FMS mode is enabled by using the **ficoncupshow fmsmode** command.

Configuring VEX_Ports

If you are going to use a VEX_Port in your tunnel configuration, use the **portCfgVEXPort** command to configure the port as a VEX_Port. VEX_Ports can be used to avoid merging fabrics over distances in FCIP implementations.

If the fabric is already connected, disable the Ethernet ports and do not enable them until after you have configured the VEX_Port. This prevents unintentional merging of the two fabrics.

VEX_Ports are described in detail in the *Fabric OS Administrator's Guide*. Refer to that publication if you intend to implement a VEX_Port.

The following example configures a VEX_Port, enables admin, and specifies fabric ID 2 and preferred domain ID 220.

```
switch:admin> portcfgvexport 18 -a 1 -f 2 -d 220
```

Configuring the media type for GbE ports 0 and 1 (Brocade 7800 switch)

Two media types are supported for GbE ports 0 and 1 on the Brocade 7800 switch: copper (RJ-45) and optical. The media type must be set for GbE ports 0 and 1 using the **portCfgGEMediatype** command. The following example configures port 1 (ge1) as an optical ports.

```
switch:admin> portcfggemediatype ge1 optical
```

The ge0 option is used for port 0 and the ge1 option is used for port 1. The copper and optical options are used for the media type.

When you enter this command without specifying the media type, the current media type for the specified GbE port is displayed, as in the following example.

```
switch:admin> portcfggemediatype ge1
Port ge1 is configured in optical mode
```

NOTE
The Optical option references the SFP bays in which optical SFPs are most often used. It is possible to insert copper-based RJ-45 SFPs into these bays.

Setting the GbE port operating mode (FX8-24 blade only)

The GbE ports on an FX8-24 blade can operate in one of three ways:

- 1 Gbps mode: GbE ports 0 through 9 may be enabled as GbE ports, with the XGE ports disabled. The 10 GbE (FTR_10G) license is not required.
- 10 Gbps mode: 10 GbE ports xge0 and xge1 may be enabled, with GbE ports 0 through 9 disabled. The 10 GbE (FTR_10G) license is required and must be assigned to the slot in which the FX8-24 blade resides.
- Dual mode: GbE ports 0 through 9 and 10 GbE port xge0 may be enabled, with xge1 disabled. The 10 GbE (FTR_10G) license is required and must be assigned to the slot in which the FX8-24 blade resides.

NOTE
Switching between 10 Gbps mode and 1 Gbps mode disrupts FCIP traffic.

NOTE
Before changing operating modes for a port, you must delete the port's FCIP configuration.

You must configure the desired GbE port mode of operation for the FX8-24 blade using the **bladeCfgGeMode --set** *mode* -slot *slot number* command. The command options are as follows.

--set mode	**1g** enables the GbE ports 0 through 9 (xge0 and xge1 are disabled).
	10g enables ports xge0 and xge1 (ge0-ge9 ports are disabled).
	dual enables the GbE ports 0 through 9 and xge0 (xge1 is disabled).
-slot *slot number*	Specifies the slot number for the FX8-24 blade.

The following example enables GbE ports 0 through 9 on an FX8-24 blade in slot 8. Ports xge0 and xge1 are disabled.

```
switch:admin> bladecfggemode --set 1g -slot 8
```

You can use the **bladecfggemode --show** command to display the GbE port mode for the FX8-24 blade in slot 8, as shown in the following example.

```
switch:admin> bladecfggemode --show -slot 8
bladeCfgGeMode: Blade in slot 8 is configured in 1GigE Mode
1GigE mode: ge0-9 ports are enabled (xge0 and xge1 are disabled)
```

Configuring port modes (Brocade 7840 switch)

You can configure the Brocade 7480 switch in either 10VE mode (default) or 20VE mode using the **extncfg** --ve-mode -10VE|20VE command. This command is disruptive as it requires rebooting the switch.

You can configure the following modes:

- **10VE mode**: In this mode 10 of the 20 total VE_Ports on the switch are enabled. A single VE_Port on a DP complex can use all Fibre Channel 20 Gbps bandwidth available to the DP complex. In 10VE mode, VE_Ports 29-33 and 39-43 are disabled.
- **20VE mode**: In this mode, all 20 VE_Ports are enabled. A single VE_Port on a DP complex can use half of the available Fibre Channel bandwidth available to the DP complex, a maximum of 10 Gbps. This option allows use of more VE_ports, but at a lower maximum bandwidth.

NOTE
10 Gbps or 40 Gbps mode is not available for the 7840 switch like 1 Gbps and 10 Gbps mode are available for the FX8-24 blade. For the 7840 switch, only configure the maximum number of VE_Ports for the 7840 switch. 10VE mode will accommodate nearly all environments and is the default.

NOTE
When switching modes, there can be no conflicting configurations or the **extncfg** command will fail. For example, if you have a tunnel on VE30, you will not be allowed to switch to 10VE mode because VE30 is disabled in that mode.

For more information on the Brocade 7840 switch port modes, refer to 10VE and 20VE port modes. on page 43

Use the following steps to configure and display the Brocade 7840 switch operating modes:

1. Connect to the switch and log in using an account assigned to the admin role.
2. Perform one of the following steps:
   - To set the operating mode to 20VE, enter the following:
     ```
     switch:admin>extncfg --ve-mode 20VE
     ```
   - To set the operating mode to 10VE, enter the following:
     ```
     switch:admin>extncfg --ve-mode 10VE
     ```
3. To display the current operating mode, enter the following:
   ```
   Switch:admin>extncfg--show
   ```
   The following displays if the switch is in 20VE mode:
   ```
   VE-Mode: configured for 20VE mode
   ```

Configuring port speed (Brocade 7840 switch)

You can configure the speed of 10 GbE ports on the Brocade 7840 switch to 1 Gbps, 10 Gbps (default), or autonegotiate using the **portCfgGe** command.

NOTE
A port set in autonegotiate mode is negotiating full duplex and pause frames (802.3X) with the attached switch. The port will not come up if there is an autonegotiate mismatch with the attached switch.

Use the following steps to configure port speed on the Brocade 7840 switch 10 GbE ports:

1. Connect to the switch and log in using an account assigned to the admin role.
2. Perform one of the following steps:
   - To set the port speed at 1 Gbps for port ge4, enter the following:
     ```
     switch:admin>portCfgGe ge4 --set -speed 1G
     ```
   - To set the port speed at 10 Gbps for port ge4, enter the following:
     ```
     switch:admin>portCfgGe ge4 --set -speed 10G
     ```
   - To set port ge4 to autonegotiate, enter the following:
     ```
     switch:admin>portCfgGe ge4 --set -speed auto
     ```
   - To disable autonegotiate on port ge4, enter the following:
     ```
     switch:admin>portCfgGe ge4 --disable -autoneg
     ```
   - To enable autonegotiate on port ge4, enter the following:
     ```
     switch:admin>portCfgGe ge4 --enable -autoneg
     ```
3. To display current port speed configuration for ge4, enter the following:
   ```
   switch:admin>portCfgGe ge4 --show
   ```

Configuring an IPIF

You must configure an IP interface (IPIF) for each circuit that you intend to configure on a Ethernet port. This is done using the **portCfg ipif create** command. The IP interface consists of an IP address, netmask, an IP MTU size, and other options depending on the extension switch or blade.

The following examples create the addressing needed for the basic sample configuration in the below figure using a Brocade 7800 switch and FX8-24 blade.

The following command creates an IP interface for port ge0 on the Brocade FX8-24 blade in slot 8 of the Brocade DCX-4S.

```
switch:admin> portcfg ipif 8/ge0 create 192.168.1.24 netmask 255.255.255.0 mtu 1500
```

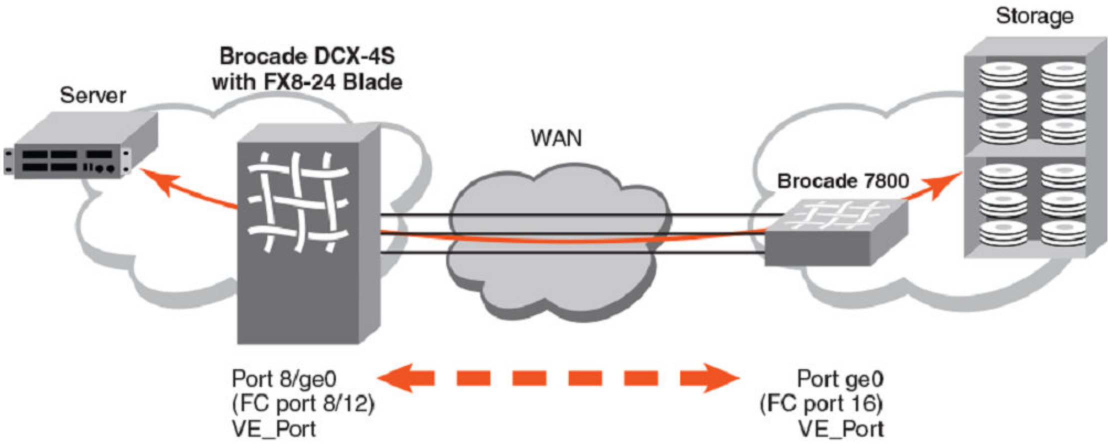
The following command creates an IP interface for port ge0 on the Brocade 7800 switch.

```
switch:admin> portcfg ipif ge0 create 192.168.1.78 netmask 255.255.255.0 mtu 1500
```

The following command displays current configuration details for all interfaces.

```
switch:admin> portshow ipif
```

FIGURE 14 Basic sample configuration



Requirements and options for configuring ipifs include the following:

- There are no addressing restrictions for IPv4 and IPv6 connections with both switches or blades in the tunnel running Fabric OS v7.0 and later.
- You can use CIDR notation for the IP4 addresses like you can for IPv6 addresses.
- You can specify an optional IP MTU size. If not specified, the size will be set to 1500 bytes.

The Brocade 7840 switch has the following additional requirements and options for configuring an ipif:

- You must assign a DP complex to the Ethernet port where commands will be received.
- For MTU the auto option can be used instead of a value which will cause any circuits using this IP address to use PTMU discovery to set the desired IP MTU.
- You must include a prefix length (pfx) or netmask.
- You can specify a VLAN ID (optional). Note that this is the only method to set the VLAN for the IP address.

The following command creates an IP interface for port ge0 on a Brocade 7840 switch. Because the Brocade 7840 switch DP0 and DP1 share Ethernet ports, port ge0 is assigned to DP0 so that DP0 receives the command. Note that a network mask (netmask), VLAN ID (vlan), and IP MTU (mtu) are specified. If CIDR is used, a prefix (pfx) is used instead of the netmask.

```
switch:admin>portcfg ipif ge0.dp0 create 192.168.0.10/24 vlan 100 mtu 1400
```

NOTE
For full details on syntax and using the **portcfg ipif** and **portshow ipif** commands, refer to the *Fabric OS Command Reference*.

Configuring an IP route

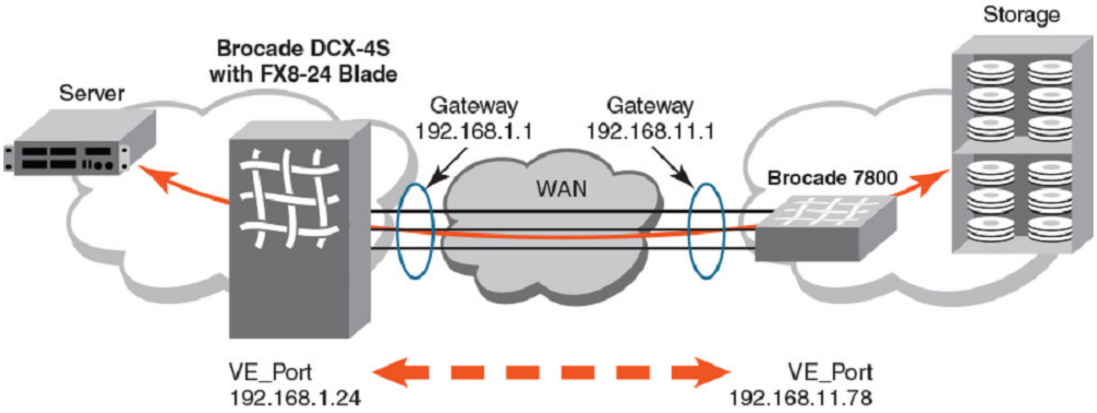
Routing is based on the destination IP address presented by an FCIP circuit. If the destination address is not on the same subnet as the Ethernet port IP address, you must configure an IP route to that destination with an IP gateway on the same subnet as the local Ethernet port IP address.

You can define up to 32 routes for each GbE port on the Brocade 7800 switch and FX8-24 blade. You can define up 128 routes per GbE port on the Brocade 7840 switch; however, you can only define 120 routes per DP. The DP limit takes precedence. Note that the **portshow iproute** command may display more routes than those you configured once all routes are added.

To configure a route, use the **portCfg iproute create** command to specify the destination IP address, subnet mask, and address for the gateway router that can route packets to the destination address. Optionally, on the Brocade FX8-24 blade, you can configure an IP route for a failover crossport using the -x or - - crossport option. For information on configuring IP routes using crossport addresses, refer to Configuring IP routes with crossports on page 34.

The following figure illustrates an IP route sample configuration.

FIGURE 15 Configuring an IP route



Commands for configuring IP routes

Following are examples of commands for configuring IP routes. You can use the same commands for Brocade FX8-24 blades, 7800 switches, and 7840 switches with modifications as noted.

The following commands are used to configure an IP route for the example configuration in the previous illustration.

- The following command creates an IP route to destination network 192.168.11.0 for port ge0 on the FX8-24 blade in slot 8 of the Brocade DCX-4S Backbone. The route is through local gateway 192.168.1.1. After the destination address, either specify a pfx (prefix length) or network mask.

  ```
  switch:admin> portcfg iproute 8/ge0 create 192.168.11.0 netmask 255.255.255.0
  192.168.1.1
  ```
- The following command creates an IP route to destination network 192.168.1.0 for port ge0 on the Brocade 7800 switch. The route is through local gateway 192.168.11.1. After the destination address, either specify a pfx (prefix length) or network mask.

  ```
  switch:admin> portcfg iproute ge0 create 192.168.1.0 netmask 255.255.255.0
  192.168.11.1
  ```
- The following command displays configured IP route information for port ge0.

  ```
  switch:admin> portshow iproute ge0
  ```

The following command creates and IP route to destination network 192.168.12.100 for port ge0 on a Brocade 7840 switch. The route is through local gateway 192.168.1.1. Because Ethernet ports are shared between DP complexes, the ge1.dp0 option directs the command to a specific DP.

```
portcfg iproute ge1.dp0 create 192.168.12.100 netmask 255.255.255.255 192.168.1.1
```

NOTE
For additional IP route configuration examples and related output from **portshow iproute** commands, refer to the *Fabric OS Command Reference*.

Commands for modifying IP routes

You can modify an existing IP route to change the local gateway address of an IP route using the **portcfg iproute** *port* **modify** command. You cannot use this command to modify the destination network address. If this needs to be modified, you must delete the IP route, and then recreate it. Also, you cannot use this command to change the prefix length or network mask.

Following is a command to change the local gateway address on a Brocade 7840 switch to 192.168.11.1. Note that for the Brocade 7840 switch only, a DP destination (in this case, dp0) is specified for the Ethernet port. Because Ethernet ports are shared between DP complexes, this option directs the command to a specific DP.

```
portcfg iproute ge1.dp0 modify 192.168.12.100 netmask 255.255.255.255 192.168.11.1
```

Validating IP connectivity

The following example tests the connectivity between the FX8-24 blade and the Brocade 7800 switch in the basic sample configuration from the Brocade 7800 switch. The **-s** option specifies the source address, and the **-d** option specifies the destination address.

```
switch:admin> portcmd --ping ge0 -s 192.168.11.78 -d 192.168.1.24
```

When using VLANS, VLAN tagging ensures that test traffic traverses the same path as real FCIP traffic. A VLAN tag entry for both the local and remote sides of the route must exist prior to issuing the **portCmd --ping** command. Refer to Managing the VLAN tag table on page 105 for details.

NOTE
To ping crossport addresses, refer to Using ping with crossports on page 35.

Creating an FCIP tunnel

Create FCIP tunnels using the **portCfg fciptunnel create** command. You configure specific tunnel options using this command, such as compression, IPsec, and FICON emulation or acceleration options. You can also specify local and remote IP addresses and circuit parameters for default circuit 0.

NOTE
When circuit options are specified on the **portcfg fciptunnel create** command and the **portcfg fciptunnel modify** command, they apply only to circuit 0. When additional circuits are added, circuit options must be applied per circuit using the **portcfg fcipcircuit create** or the **portcfg fcipcircuit modify** command.

A suggested technique is to configure the tunnel with appropriate tunnel parameters only (no IP addresses or circuit options). This may be useful in staging a configuration without committing specific circuit parameters. Then you can configure circuit 0 and additional circuits using **portcfg fcipcircuit** commands.

NOTE
A Brocade 7840 switch can only connect with another Brocade 7840 switch through FCIP. It cannot connect to a Brocade 7500 switch, 7800 switch, or FX8-24 blade.

Creating an FX8-24 and Brocade 7800 tunnel

NOTE
You cannot create a tunnel from a Brocade 7840 switch to a Brocade 7800 switch or FX8-24 blade.

To create an FX8-24 tunnel endpoint using the **portcfg fciptunnel** command, VE_Port 12 is specified on slot 8. Circuit 0 is created automatically when the FCIP tunnel is created. An FCIP tunnel is represented by a VE_Port. The Brocade 7800 switch remote or destination address (192.168.11.78) is specified first, followed by the FX8-24 local or source address (192.168.1.24). ARL minimum (-b) and maximum (-B) committed rates are specified for circuit 0.

```
switch:admin> portcfg fciptunnel 8/12 create --remote-ip 192.168.11.78 --local-ip
192.168.1.24 -b 12000 -B 1000000
```

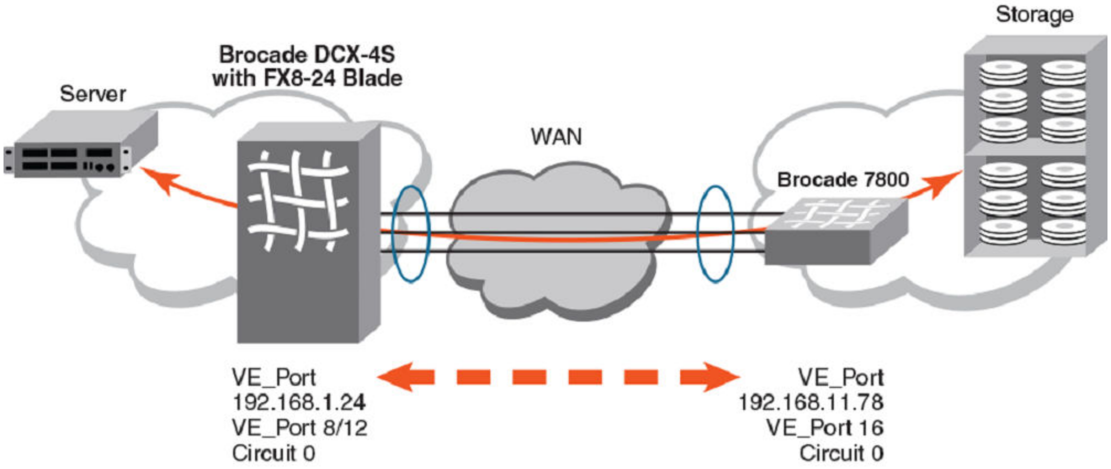
The following command creates the Brocade 7800 tunnel endpoint. VE_Port 16 is specified. Circuit parameters are included to create circuit 0 on the Brocade 7800 switch. The circuit parameters must match up correctly with the circuit parameters on the FX8-24 end of the circuit. The FX8-24 remote or destination address is specified first (192.168.1.24), followed by the 7800 switch local or source address (192.168.11.78). Matching ARL minimum and maximum committed rates must be specified on both ends of circuit 0.

```
switch:admin> portcfg fciptunnel 16 create --remote-ip 192.168.1.24 --local-ip
192.168.11.78 b 500000 -B 1000000
```

For a description of circuit and tunnel configuration options that you can include on the **portcfg fciptunnel** command, refer to Tunnel configuration options on page 70.

The following figure illustrates the results of the configuration.

FIGURE 16 Adding an FCIP tunnel to the basic sample configuration



Creating Brocade 7840 tunnels

Tunnel configuration for Brocade 7840 switches is similar to Brocade 7800 and FX8-24 switches, but have some different tunnel options.

You can only create a tunnel from a Brocade 7840 switch to another Brocade 7840 switch. You cannot create tunnels between Brocade 7840 switches and Brocade 7500 switches, 7800 switches, or FX8-24 blades.

The following example configures a tunnel endpoint on a Brocade 7840 switch at VE_Port 25.

```
switch:admin> portcfg fciptunnel 25 create --local-ip 192.168.2.15 --remote-ip
192.168.2.25 -b 500000 -B 10000000 -c deflate --ipsec policy1
```

- To create a Brocade 7840 switch endpoint using the **portcfg fciptunnel** command, VE_Port 25 is specified.
- The circuit 0 local or source address (192.168.2.15) is specified, followed by the remote or destination address (192.168.2.25).
- ARL minimum (-b) and maximum (-B) committed rates are specified for circuit 0. These must match the rates configured on the remote 7840 switch.
- Deflate compression is enabled.
- IPsec is enabled using policy1

The following command creates the destination Brocade 7840 switch endpoint. VE_Port 34 is specified.

```
switch:admin> portcfg fciptunnel 34 create --local-ip 192.168.2.25 --remote-ip
192.168.2.15 -b 500000 -B 1000000 -c deflate --ipsec policy1
```

For the destination switch, the same circuit and tunnel options are configured as the other Brocade 7840 switch. The circuit options for circuit 0, such as ARL minimum and maximum committed rates, must match.

For a description of circuit and tunnel configuration options that you can include on the **portcfg fciptunnel** command, refer to Tunnel configuration options on page 70.

Tunnel configuration options

The following tables detail tunnel and circuit configuration options available on the **portcfg fciptunnel** command.

Most FCIP features are enabled using optional arguments available on the **portcfg fciptunnel create** command and the **portcfg fciptunnel modify** command. Some of these arguments apply only to FCIP tunnels, and are used only on the **portcfg fciptunnel create** command and the **portcfg fciptunnel modify** commands.

TABLE 14 Tunnel options

Option	Arguments	Disruptive	Description
Compression	Short option: **-c** Long option: **--compression** Operands 7800 and FX8-24: **0** |**none**|**hardware**|**moderate** |**aggressive**|**auto** Operands 7840: **none**|**deflate**|**agr-deflate**|**fast-deflate**|	Yes	For all extension products, enables compression on an FCIP tunnel. Compression is set by the **portCfg fciptunnel create or modify** command, and applies to traffic over all circuits in the tunnel. Compression cannot be set or modified by the **portCfg fcipcircuit create** or **portCfg fcipcircuit modify** This feature is available on the 7800 switch and FX8-24 blade only. The argument values have the following meanings: • None - Disables compression • Hardware - Enables Standard compression mode • Moderate - Enables Moderate compression mode • Aggressive - Enables Aggressive compression mode • Auto - Enables Auto compression mode This feature is available on the Brocade 7840 switch only. The argument values have the following meanings. • None - No compression. • Deflate - Enables deflate compression mode. • aggr-def - Enables aggressive deflate compression mode. • fast-deflate - Enables fast-deflate compression mode. For a description of the compression modes refer to Compression options on page 18.

TABLE 14 Tunnel options (Continued)

Option	Arguments	Disruptive	Description
FCIP Fastwrite	Short option: **-f** Long option: **--fast-write** Operands (modify only): **enable** |**disable** • Create behavior: No operands required. FCIP Fastwrite enabled If specified on create. • Modify behavior: Requires operands.	Yes	This feature is available on all extension products. Enables or disables FCIP Fastwrite. FCIP Fastwrite is initially disabled, and must be enabled to take effect.
No Read Pipelining	Short option:**-N** Long option: --**-no-read-pipelining**	Yes	This feature is available on the Brocade 7800 switch and FX8-24 blade only. Disables FCP Tape Read Pipelining
OSTP	Short option: **-t** Long option: **--tape-pipelining** Operands (modify only): **enable** |**disable| write-only** • Create behavior: Operands not required. OSTP enabled when specified on create. • Modify behavior: Requires operands.	Yes	For all extension products, disables or enables tape Open Systems Tape Pipelining (OSTP). OSTP is initially disabled. Both FCIP Fastwrite and OSTP must be enabled if you want to implement OSTP, as described in FastWrite and Open Systems Tape Pipelining on page 19. The argument values have the following meanings. disable - OSTP Disabled enable - OSTP Read/Write Enabled write-only - OSTP Write Enabled

TABLE 14 Tunnel options (Continued)

Option	Arguments	Disruptive	Description
QoS Priority Percentages	Short option**-q** high *percentage*, med *percentage*, low *percentage* Long option: **--qos** high *percentage*, med *percentage*, low *percentage*, or **--qos-high** *percentage*, **--qos-med** *percentage*, **--qos-low** *percentage* Operands: *percentage*. Whole values from 10-80 (total 100).	Yes	This feature is available on all extension products. This sets Quality of Service (QoS) priority percentages to default values 50% for high, 30% for medium, and 20% for low. You can change percentages with the *percentage* option. Priorities are enforced only when there is congestion on the network. If there is no congestion, all traffic is handled at the same priority. Note that IPsec is the best-practice method of security that will prevent all attacks.
Remote FC WWN	Short Option: **-n** Long Option: **--remote-wwn** Operands: *remote-wwn* - specifies the WWN of the remote FC entity.	Yes	This is a fabric security feature for all extension products that allows the FCIP tunnel to come up only when the correct remote WWN is entered. If the WWN of the remote side does not match the value entered here, the FCIP tunnel will not initiate.
Enable IPsec	Short Option: **-I** Long Option: **--ipsec** Operands (modify only): **enable** |**disable** • Create behavior: Operands are not required. IPsec enabled when specified on create. • Modify behavior: Requires operands.	Yes	This feature is available on the Brocade 7800 switch and FX8-24 blade only. Disables or enables IPsec on this FCIP tunnel. Refer to Implementing IPsec over FCIP tunnels on page 106 for information about IPsec policies.
Enable IPsec policy	Short option: **-I**. Long option: **--ipsec** Operands: *policy*|**none**.	Yes	This feature is available on the Brocade 7840 switch only. Enables the IPsec policy for the tunnel or disables IPsec with none option. The *policy* is the policy name configured with the **portCfg ipsec-policy** command.

TABLE 14 Tunnel options (Continued)

Option	Arguments	Disruptive	Description
Legacy IPsec connection	Short Option: **-l** Long Option: **--legacy** Operands (modify only): **enable** |**disable** • Create behavior: Operands are not required. Legacy IPsec enabled when specified on create. • Modify behavior: Requires operands.	Yes	This feature is available on the Brocade 7800 switch and FX8-24 blade only. Enables or disables legacy IPsec mode. Legacy IPsec uses the IPsec connection process compatible with Fabric OS versions prior to v7.0.0.
IKE V2 authentication Key for IPsec	Short Option: **-K** Long Option: **--key** Operands (modify and create): **key**	Yes	This feature is only available on the Brocade 7800 switch and FX8-24 blade. This is the pre-shared key used during IKE authentication. The key must be 32 characters in length. **NOTE** For the Brocade 7840 switch, the IKE V2 authentication key is provided using the **portCfg ipsec-policy** command.
FICON mode	Short Option: **-F** Long Option: **--ficon** Operands (modify only): **enable** |**disable** • Create behavior: Operands are not required. FICON mode enabled when specified on create. • Modify behavior: Requires operands.	Yes	Disables or enables FICON mode. The **-F** option performs the following tasks: • Changes the default circuit keep-alive timeout value (KATOV) to 1 second when the next circuit is created for this tunnel. • Allows you to enable the Advanced FICON Acceleration (AFA) features on the tunnel if the AFA license is applied to the switch.

TABLE 15 Circuit options

Option	Argument	Disruptive	Description
Committed rate	*committed rate* Create behavior: Sets the minimum and maximum committed rate to the value specified for *committed rate* . Short option: **-b** and**-B**	7800 and FX8-24 - Yes 7840 - No	This option may be used on a **portcfg fciptunnel create** command or the **portcfg fcipcircuit create** command to set a committed rate for an FCIP circuit. When this option is used with the **portcfg fciptunnel create** command, the committed rate applies only to circuit 0. To modify the committed rate or if you intend to use ARL on the circuit, use the -b to set the minimum committed rate and the -B options to set the maximum committed rate. If ARL is not going to be used, you can use a single data rate. If ARL is going to be used, you must use the -b and -B options to define minimum and maximum rates. Values entered for -b and -B can be the same and therefore describe a configuration with a fixed data rate.

TABLE 15 Circuit options (Continued)

Option	Argument	Disruptive	Description
Adaptive Rate Limiting (ARL)	Short option: **-b** Long option: **--min-comm-rate** Operands: *kbps*	7800 and FX8-24 - Yes 7840 - No	**NOTE** This information pertains to the ARL feature. The minimum committed rate is a guaranteed minimum traffic rate for an FCIP circuit. For the Brocade 7800 switch and FX8-24 blade, the valid ranges for **-min-comm-rate** are 10,000 Kbps through 1,000,000 Kbps for 1 GbE ports and 10,000 Kbps to 10,000,000 Kbps for 10 GbE (XGE). For the Brocade 7840 switch, valid range for **-min-comm-rate** are 20,000 Kbps through 10,000,000 Kbps for 10 GbE or 40 GbE ports. **NOTE** When added together, the minimum committed rates for all circuits cannot exceed the speed of the 1 GbE, 10 GbE (XGE), or 40 GbE ports. The rate does not need to be an integral value of 1,000,000, and both sides of the tunnel must have matching configurations.

TABLE 15 Circuit options (Continued)

Option	Argument	Disruptive	Description
	Short option: **-B** Long option: **--max-comm-rate** Operands: **kbps**	7800 and FX8-24 - Yes 7840 - No	The maximum committed rate is the rate that the tunnel will try to achieve, based on availability and network performance. The valid ranges for **-max-comm-rate** are 10,000 Kbps to 1,000,000 Kbps for 1 GbE ports and 10,000 Kbps to 10,000,000 Kbps for 10 GbE ports. For the Brocade 7840 switch, this range is 10,000 Kbps to 10,000,000 Kbps for 10 GbE ports **NOTE** When ARL is used, the link cost is equal to the sum of the maximum traffic rates of all established and currently active lowest metric circuits in the tunnel. The rate does not need to be an integral value of 1,000,000, the maximum committed rate can be no larger than five times the minimum committed rate, and both sides of the tunnel must have matching configurations.

TABLE 15 Circuit options (Continued)

Option	Argument	Disruptive	Description
ARL algorithm mode (Brocade 7840 switch only)	Short option: **-A** Long option **--arl-algorithm** Operands: **auto**|**reset**|**step-down**|**timed-step-down**	Yes	This feature is only available on the Brocade 7840 switch. Sets the ARL algorithm. Operands are auto|reset|step-down|timed-step-down. • Auto - Let ARL determine the best method. • Reset - All connections reset to minimum rate. For use with less error tolerant devices. • Step-down - Connections step down incrementally. For use with shorter links with error tolerant devices. • Timed-step-down - Connections step down incrementally on specific time slices. For use with long latency links.
Selective Acknowledgment	Short option: **-s** Long option: -**-sack** Operands (modify only): **enable|disable** • Create behavior: Operands are not required. Selective acknowledgment will be disabled when specified on create. • Modify behavior: Requires operands.	Yes	This feature is only available on the Brocade 7800 switch and FX8-24 blade. Disables or enables selective Acknowledgment. Selective Acknowledgment allows a receiver to acknowledge multiple lost packets with a single ACK response. This results in better performance and faster recovery time. Selective Acknowledgment is initially turned on. For some applications and in some situations, you may need to turn selective Acknowledgment off. This option is used to toggle the option off and on.

TABLE 15 Circuit options (Continued)

Option	Argument	Disruptive	Description
Keep-alive timeout	Short option: **-k** Long option: **--keepalive-timeout** Operands: **ms**	Yes	Specifies the keep-alive timeout value (KATOV) in milliseconds. The valid range is 500 ms (.5 seconds) to 720000 ms (720 seconds). If the tunnel has circuits enabled for FICON emulation, circuits will default to 1000 ms (1 second). If FICON emulation is not enabled, then circuits will be created with 10000 ms (10 seconds) for the keep-alive timeout.
Minimum retransmit time	Short option: **-m** Long option: **--min-retrans-time** Operands: **ms**	No	This feature is only available on the Brocade 7800 switch and FX8-24 blade only. The minimum retransmit time, in milliseconds. The range of valid values is 20 through 5,000 ms and the default is 100 ms. As a best practice, do not alter this value unless instructed to do so by technical support.
Failover/standby metric	Short option: **-x** Long option: **--metric** Operands: **0** | **1**	Yes	You can configure standby circuits by assigning a metric 0 or 1. A lower metric assigns a higher priority to the circuit. As data is flowing through the FCIP tunnel, it automatically traverses the lowest metric circuits. Refer to Circuit failover on page 51 for a description of circuit failover and the use of standby circuits.

TABLE 15 Circuit options (Continued)

Option	Argument	Disruptive	Description
VLAN Tagging	**Short option: -v** **Long option: --vlan-tagging** **Operands: vlan-id**	Yes	This feature is only available on the Brocade 7800 switch and FX8-24 blade only. Applies VLAN tagging to a circuit and sets a specific Layer 2 Class of Service (CoS). Specify a *vlan_id*. Valid values are from 1 through 4095 and the default is 1. Refer to Managing QoS, DSCP, and VLANs on page 102 for information about VLAN tagging.
Class of Service (CoS)	Class of Service options (use with VLAN tagging options and operand): **--l2cos-f-class n** **--l2cos-high n** **--l2cos-medium n** **--l2cos-low n**	No	Sets the Layer 2 Class of Service (L2CoS) options for VLAN tagging. Options are for F-Class traffic, and high, medium, and low priority traffic. Specify a value for *n* from 0 through 7 (default is 0). **NOTE** L2CoS is also known as 802.1P which is a part of 802.1Q VLAN Tagging. 802.1P requires the use of 802.1Q.
DSCP Tagging	DSCP tag options (use with VLAN tagging options and operand): **- -dscp-f-class n** **- -dscp-high n** **- -dscp-medium n** **- -dscp-low n**	No	Applies a DSCP tag to a circuit. Specify a value for *n* from 0 through 63 (default is 0). Refer to Managing QoS, DSCP, and VLANs on page 102 for information about DSCP tagging.

TABLE 15 Circuit options (Continued)

Option	Argument	Disruptive	Description
Specify connection type	Short option: **-C** Long option: **--connection-type** Operands: **default** | **listener** |**initiator**	Yes	Allows you to specify which side of the circuit is the listener or initiator. If this is not specified, the initiator or listener is automatically selected based on the lower- and higher-order IP address. In NAT environments, this can cause problems as both sides of the circuit may have lower-order IP addresses. When setting initiator or listener options, a firmware download to a previous version will not be allowed until you set the default option.
Maximum retransmits	Short option: **-r** Long option: **--max-retransmits** Operands: **rtx**	No	This feature is only available on the Brocade 7800 switch and FX8-24 blade only. Sets the maximum number of retransmits for the FCIP circuit before the connection will be brought down. If operating on a lossy network, increasing this value may allow the FCIP circuit to remain active when it may otherwise fail. Specify a value for *rtx* from 1 through 16 (default is 8).
Administrative status	Short option: **-a** Long option: **--admin-status** Operands (modify and create): **enable** | **disable**	Yes	Disables or enables the FCIP circuit. Administrative Status is disabled by default.

Keep-alive timeout option

Consider the following items when configuring the keep-alive timeout value (KATOV):

- A FICON tunnel requires a KATOV of less than or equal to 1 second for each FCIP circuit added to a tunnel.
- If the tunnel is created first with the FICON flag, then the KATOV for all added circuits will be 1 second (recommended value for FICON configurations).
- If the tunnel is created with one or more circuits, and the tunnel is modified to be a FICON tunnel, then the circuits that were previously created must be modified to have the correct KATOV.
- Set the FCIP circuit KATOV to the same value on both ends of an FCIP tunnel. If local and remote circuit configurations do not match, the tunnel will use the lower of the configured values.
- For normal operations over FCIP tunnels, the KATOV for all FCIP circuits in an FCIP tunnel must be less than the overall I/O timeout for all FC exchanges. If the FC I/O timeout value is less than the KATOV, then inputs and outputs will time out over all available FCIP circuits without being retried.

The KATOV should be based on application requirements. Check with your FC initiator providers to determine the appropriate KATOV for your application. The sum of KATOVs for all circuits in a tunnel should be close to the overall FC initiator I/O timeout value. As an example, a mirroring application has a 6-second I/O timeout. There are three circuits in the FCIP tunnel. Set the KATOV to 2 seconds on each FCIP circuit. This will allow for maximum retries over all available FCIP circuits before an I/O is timed out by the initiator.

Refer to the keep-alive timeout option in Tunnel configuration options on page 70 for information on option format and value range.

Creating additional FCIP circuits

Additional FCIP circuits can be created and added to an FCIP tunnel (VE_Port) using the **portCfg fcipcircuit create** command. The following examples add a circuit to the tunnel in the basic sample configuration illustrated in Creating an FCIP tunnel on page 68. Note that although these examples use a FX8-24 blade and a Brocade 7800 switch, examples for the Brocade 7840 switch would be similar.

NOTE
For the 7800 switch and FX8-24 blade, you must enable the Advanced Extension (FTR_AE) license to add circuits. This license is enabled on the 7840 switch when shipped from the factory.

The following command creates circuit 1 on the FX8-24 end of the trunk.

```
switch:admin> portcfg fcipcircuit 8/12 create 1 --remote-ip 192.168.11.78 --local-ip
192.168.1.25 -b 15500 -B 62000
```

The following command displays configuration details for circuit 1.

```
switch:admin> portshow fcipcircuit 8/12 1
```

The following command creates circuit 1 on the Brocade 7800 switch end of the trunk.

```
 switch:admin> portcfg fcipcircuit 16 create 1 --remote-ip 192.168.1.25 --local-ip
192.168.11.78 -b 15500 -B 62000
```

The following command displays configuration details for circuit 1.

```
switch:admin> portshow fcipcircuit 1
```

Note the following information about the basic sample configuration:

- The VE_Ports used to create the tunnel are the same as those specified on the FCIP tunnel in the basic sample configuration. The VE_Ports uniquely identify the trunk, and the circuit is associated with this specific trunk.
- The unique destination and source IP addresses are mirrored on either end of the tunnel. The address 192.168.11.78 is the destination address for the FX8-24 blade, and the source address for the Brocade 7800 switch, while the address 192.168.1.25 is the destination address for the Brocade 7800 switch, and the source address for the FX8-24 blade.

Also note the following about configuring circuits in general:

- ARL minimum and maximum rates are set per circuit. They must be the same on either end of a circuit, but individual circuits may have different rates.
- You can configure standby circuits to operate during circuit failover by assigning a metric. In the following example, circuit 2 is used only when circuit 1 fails.

```
switch:admin> portcfg fcipcircuit 8/12 create 1 --remote-ip 192.168.11.78 --local-
ip 192.168.1.25 -b 155000 -B 620000
switch:admin> portcfg fcipcircuit 8/12 create 2 --remote-ip 192.168.11.8 --local-
ip 192.168.1.26 -b 155000 -B 620000 -x 1
```

- When multiple FCIP tunnels are present on a switch and additional circuits are added to an active tunnel, some frame loss can occur for a short period of time because the internal Fibre Channel frame routing tables in the switch are refreshing. Therefore, add additional circuits only during low I/O periods on the FCIP trunk being modified. In addition, if deleting or adding a circuit increases or decreases the total trunk bandwidth, then disable and re-enable the tunnel (VE_Port) after deleting or adding the circuit. This will allow the switch to adjust internal routes to fully utilize the new bandwidth. FCIP tunnels and trunks are designed to produce FSPF costs that make them unlikely to be the preferred path to reach all destinations within the local site. This prevents local traffic from taking a long path across the FCIP connection.

Verifying the FCIP tunnel configuration

After you have created local and remote FCIP configurations, verify that the FCIP tunnel and circuit parameters are correct using the **portshow fciptunnel** command. Refer to the *Fabric OS Command Reference* for a description of the command syntax and output.

Configuring Extension HCL

Extension HCL is an optional configuration for the Brocade 7840 Extension Switch. The only configuration that you must provide for Extension HCL are additional IP addresses for the local backup tunnel (LBT) and remote backup tunnel (RBT) endpoints. There are no subnet restrictions for configuring the IP addresses for these endpoints. The IP network must be capable of delivering traffic to all these IP address from any IP address. The circuits of the primary tunnel (PT) or trunk are automatically replicated on the LBT and RBT, including the circuit's properties such as QoS markings, FastWrite, and FICON Acceleration. This maintains the FCIP environment during the Extension HCL process. For more information on the Extension HCL feature, refer to Extension Hot Code Load on page 44.

A LBT that is protecting a PT will stay online during the Extension HCL process. The Brocade 7840 can be configured to operate either in 10VE or 20VE mode for Extension HCL. Each DP on 10VE mode can accommodate 5 PTs and 5 LBTs. If 5 protected tunnels were configured, there would be a maximum of 15 tunnels (5 PTs, 5 LBTs, and 5 RBTs). Each DP in 20VE mode can accommodate 10 PTs and 10 LBTs. If 10 protected tunnels were configured, there would be a maximum of 30 tunnels (10 PTs, 10 LBTs, and 10 RBTs).

A PT must contain at least one protected circuit. To configure a protected circuit, you must provide two additional IP addresses for the circuit. Not all circuits must be protected by Extension HCL. Only the circuits that are configured with the additional Extension HCL IP addresses will stay up during firmware updates. The non-protected circuits will go down. The FCIP trunk would run at a diminished capacity if there were not a one-for-one protection of the circuits resulting in a lesser aggregate bandwidth..

To provide the additional IP addresses, use the --local-ha-ip and --remote-ha-ip options in the **portcfg fcipcircuit create** command. You can also use these options in the **portcfg fciptunnel create** command if you wish to create circuit 0 using that command. Following is an example of creating a protected circuit for a tunnel.

```
portcfg fcipcircuit <ve_port> create <circuit_ID> --local-ip <ipaddr> --remote-ip
<ipaddr> --local-ha-ip <ipaddr> --remote-ha-ip <ipaddr>
```

For detailed instructions to create backup tunnels for the Extension HCL feature, refer to Configuring backup tunnels on page 84.

Once you configure backup tunnels, you can monitor Extension HCL status on tunnels during the firmware download using the **portshow fciptunnel** --hc -status command as in the following example. Note that tunnels 24, 25, and 26 are protected tunnels.

```
portshow fciptunnel --hc -status
Checking FCIP Tunnel HA Status.
   Current Status       : Ready  *System Status*
   CP Version           : v7.3.0
DP0 Status:
   State             : Online – Inactive  *DP0 Status Online and HCL inactive on DP0*
   Current Version   : v7.3.0
   Current HA Stage  : INITIAL   *Current HCL stage on DP0, will change when HCL
active on DP0*
DP1 Status:
   State             : Online - Inactive
   Current Version   : v7.3.0
   Current HA Stage  : INITIAL
 Tunnel 24 is configured for HA and is HA Online. Traffic will not be disrupted.
*List of all VE and states*
 Tunnel 25 is configured for HA and is HA Online. Traffic will not be disrupted.
 Tunnel 26 is configured for HA and is HA Online. Traffic will not be disrupted.
```

Configuring backup tunnels

A backup tunnel is required for each DP on a Brocade 7840 switch to support the Extension hot code load (HCL) feature. Configure these tunnels, along with the PT, using the **portcfg fciptunnel** and **portcfg fcipcircuit** commands.

A backup tunnel contains at least one circuit with local (--local-ha-ip) and remote (--remote-ha-ip) backup IP addresses. These addresses support the Extension HCL feature by enabling the PT traffic to fail over to the DP that is not currently being upgraded with new firmware so that this traffic is not disrupted during the upgrade process. These addresses and the circuits they support are "protected" from going down during a firmware upgrade.

LBT and RBT are created automatically during the Extension HCL process if the Extension HCL IP addresses are configured. There are no restrictions on the subnet to which the IP addresses belong. The circuit properties such as VLAN, QoS markings, FastWrite, and FICON Acceleration are maintained on the backup tunnel.

An example for creating local and remote IP addresses using the **portCfg fcipcircuit create** command follows:

```
portcfg fcipcircuit 24 create 1 --local-ip 192.168.2.15 --remote-ip
192.168.11.78 --local-ha-ip 192.168.2.31 --remote-ha-ip 192.168.11.68 -b 155000 -B
620000
```

The --local-ip and --remote-ip options provide the PT IP addresses. The --local-ha-ip and --remote-ha-ip options provide the LBT and RBT IP addresses respectively.

The following steps illustrate how to configure two circuits (0 and 1) for ge2, each with Extension HCL IP addresses, for an FCIP tunnel from source VE_Port 24.

1. Configure ipifs that you will use for circuit 0 as in the following example.

   NOTE
   The local interface (Ethernet port) assigned to the LBT IP address must be assigned to the DP other than the DP where the local PT IP address is assigned. Because the Brocade 7840 switch shares Ethernet ports, you can assign an Ethernet port to a specific DP using the **portcfg ipif** command with the ge_port.dp_num option.

   ```
   switch:admin> portcfg ipif ge2.dp0 create 192.168.2.10 netmask 255.255.255.0 mtu
   1500
   switch:admin> portcfg ipif ge2.dp1 create 192.168.2.30 netmask 255.255.255.0 mtu
   1500
   ```

   The ge2.dp0 option assigns port ge2 to DP0 and ge2.dp1 assigns ge2 to DP1. The address assigned to ge2 on DP0 will be for the local PT, while the address assigned to ge2 on DP1 will be the LBT.

2. Configure ipifs that you will use for circuit 1 in the same fashion, as in the following example:

   ```
   switch:admin> portcfg ipif ge2.dp0 create 192.168.2.15 netmask 255.255.255.0 mtu
   1500
   switch:admin> portcfg ipif ge2.dp1 create 192.168.2.31 netmask 255.255.255.0 mtu
   1500
   ```

3. Configure a tunnel for VE_24 and circuit 0 for the tunnel using the following example.

   ```
   switch:admin> portcfg fciptunnel 24 create --local-ip 192.168.2.10 --remote-ip
   192.168.2.20 --local-ha-ip 192.168.2.30 --remote-ha-ip 192.168.2.40 -b 12000 -B
   1000000
   ```

4. Configure circuit 1 for the tunnel using the following example:

   ```
   switch:admin> portcfg fcipcircuit 24 create 1 --local-ip 192.168.2.15 --remote-ip
   192.168.2.25 --local-ha-ip 192.168.2.31 --remote-ha-ip 192.168.2.41 -b 15500 -B
   62000
   ```

5. Verify the tunnel configuration using the **portshow fciptunnel** -hc command as in the following example:

```
switch:admin> portshow fciptunnel --circuit -hc

Tunnel Circuit  OpStatus  Flags    Uptime  TxMBps  RxMBps ConnCnt CommRt Met/G
-----------------------------------------------------------------------------
25    -         Up      -M-fTF-a 2h21m40s   21.03   18.52    1       -      -/-
25    0 ge2     Up      ---rh--4 2h21m40s   10.51    9.27    1  4000/5000  0/-
25    1 ge2     Up      ---rh--4 2h21m36s   10.77    9.25    1  4000/5000  0/-
25    -         Up      -R-fTF-a 2h21m39s    0.00    0.00    1       -      -/-
25    0 ge2     Up      ---rh--4 2h21m40s    0.00    0.00    1  4000/5000  0/-
25    1 ge2     Up      ---rh--4 2h21m37s    0.00    0.00    1  4000/5000  0/-
25    -         Up      -L-fTF-a 2h21m43s    0.00    0.00    1       -      -/-
25    0 ge2     Up      ---rh--4 2h21m43s    0.00    0.00    1  4000/5000  0/-
25    1 ge2     Up      ---rh--4 2h21m41s    0.00    0.00    1  4000/5000  0/-
-----------------------------------------------------------------------------
 Flags (tunnel): M=MainTunnel  L=LocalBackup  R=RemoteBackup
                 i=IPSec f=Fastwrite T=TapePipelining F=FICON r=ReservedBW
                 A=AdvCompr a=FastDeflate d=Deflate D=AggrDeflate
      (circuit): h=HA-Configured v=VLAN-Tagged p=PMTU 4=IPv4 6=IPv6
                 ARL a=Auto r=Reset s=StepDown t=TimedStepDown
```

Enabling persistently disabled ports

Use the following steps to enable persistently disabled ports.

1. Connect to the switch and log in using an account assigned to the admin role.
2. Enter the **portCfgShow** command to view ports that are persistently disabled.

3. After identifying the ports, enter the **portCfgPersistentEnable** command to enable the ports.
4. Enter the **portCfgShow** command to verify the port is persistently enabled.

Disabling ports with FMS Mode enabled

If you enter **portCfgPersistentEnable** and receive "command not allowed in fmsmode" or "command not found" messages, FMS mode may be enabled. You cannot use the **portCfgPersistentEnable** or **portCfgPersistentDisable** commands with FMS mode enabled. Use the **portEnable** and **portDisable** commands instead.

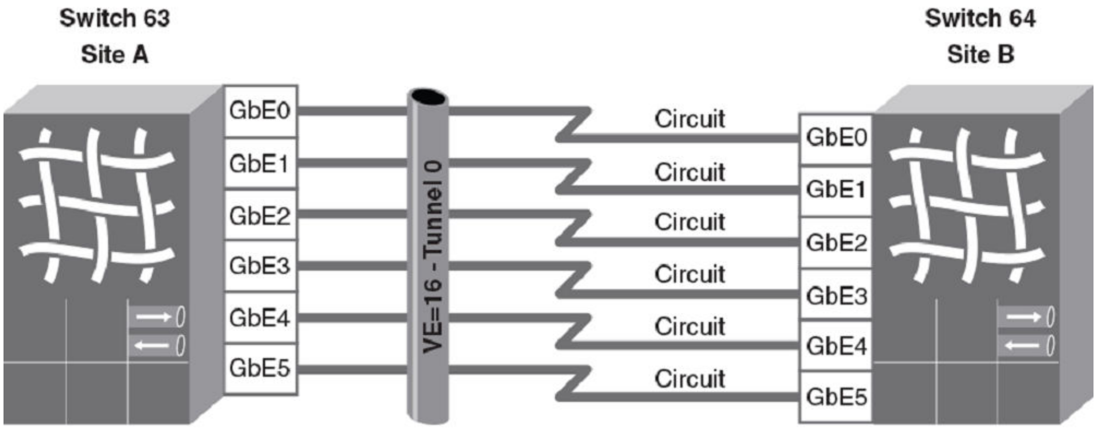
You can determine if FMS mode is enabled by using the **ficoncupshow fmsmode** command.

NOTE
The **portCfgPersistence** command also cannot be used with FMS mode enabled. This command sets or removes the persistent disable flag on a port or range of ports. If FMS mode is enabled, use the **portdisable** command with Active=Saved mode enabled instead.

Creating an FCIP trunk (example)

This section provides procedures and applicable commands to create a tunnel containing six circuits between two switches or blades. The figure below illustrates an example of these circuits between two FX8-24 blades inside a DCX chassis.

FIGURE 17 FCIP trunk example with six circuits



Six circuits per trunk are configured in the example. For specifications on the number of circuits allowed per trunk and interface on the Brocade 7800 switch, Brocade 7840 switch, and FX8-24 blade, refer to the heading for your extension product under Tunnel and circuit requirements on page 47.

The following examples use a Brocade FX8-24 blade and the Brocade 7800 switch. General procedures for the Brocade 7840 switch are the same, but there are differences in command requirements. For more details on these differences and on the procedures and commands used in this example in general, refer to the following sections:

- Configuring an IPIF on page 65
- Configuring an IP route on page 66

- Creating an FCIP tunnel on page 68
- Creating additional FCIP circuits on page 82

To create a tunnel or trunk between two switches or blades, you must first understand the IP network infrastructure between the sites. Each circuit requires a pair of IP interface addresses (either IPv4 or IPv6). Therefore, to create an FCIP trunk with six circuits, you need 12 IP addresses: six for the site A switch and six for the site B switch. In the simplest configuration (non-routed), the IP addresses can all be on the same IP subnet with no IP routes. In routed configurations, you will also define IP routes (IP gateway addresses). In the following example, six different IP subnets are used, although this is not a requirement.

The following steps create the multicircuit tunnel (FCIP trunk) example illustrated in the preceding figure.

1. Assign IP addresses to each switch Ethernet port using the **portcfg ipif** command. The **portcfg ipif** command requires the IP address, subnet mask, and the IP MTU for that IP interface. The following examples show how to create the IP interfaces (IPIFs) for this configuration.

   Site A

   ```
   portcfg ipif ge0 create 192.168.0.63 netmask 255.255.255.0 mtu 1500
   portcfg ipif ge1 create 192.168.1.63 netmask 255.255.255.0 mtu 1500
   portcfg ipif ge2 create 192.168.2.63 netmask 255.255.255.0 mtu 1500
   portcfg ipif ge3 create 192.168.3.63 netmask 255.255.255.0 mtu 1500
   portcfg ipif ge4 create 192.168.4.63 netmask 255.255.255.0 mtu 1500
   portcfg ipif ge5 create 192.168.5.63 netmask 255.255.255.0 mtu 1500
   ```

   Site B

   ```
   portcfg ipif ge0 create 192.168.0.64 netmask 255.255.255.0 mtu 1500
   portcfg ipif ge1 create 192.168.1.64 netmask 255.255.255.0 mtu 1500
   portcfg ipif ge2 create 192.168.2.64 netmask 255.255.255.0 mtu 1500
   portcfg ipif ge3 create 192.168.3.64 netmask 255.255.255.0 mtu 1500
   portcfg ipif ge4 create 192.168.4.64 netmask 255.255.255.0 mtu 1500
   portcfg ipif ge5 create 192.168.5.64 netmask 255.255.255.0 mtu 1500
   ```

2. Create the FCIP tunnel using the **portcfg fciptunnel** command. The following example creates an empty tunnel with hardware compression enabled to which you will add circuits. The FCIP tunnels are represented in the switch as VE_Ports. There are several ways to create this tunnel as shown by the following options:

   - To create the tunnel with hardware compression, use the following commands.

     Site A

     ```
     portcfg fciptunnel 16 create -c 1
     ```

     Site B

     ```
     portcfg fciptunnel 16 create -c 1
     ```

   - To use this tunnel for FICON traffic with hardware compression, create it with the following commands.

     Site A

     ```
     portcfg fciptunnel 16 create --ficon -c 1
     ```

     Site B

     ```
     portcfg fciptunnel 16 create --ficon -c 1
     ```

   - To use this tunnel for FCP with Fastwrite and Open Systems Tape Pipelining traffic, and hardware compression, create it using the following commands.

     Site A

     ```
     portcfg fciptunnel 16 create --fastwrite --tape-pipelining -c 1
     ```

Site B

```
portcfg fciptunnel 16 create --fastwrite --tape-pipelining -c 1
```

NOTE
For the Brocade 7840 switch, compression options for compression (-c) are specified as none, deflatee, aggr-def, and faset-deflate. Refer to Tunnel configuration options on page 70.

To display details of the tunnel configuration after using the previous commands, use the **portshow fciptunnel all** command.

At this time, the tunnel has been created with the compression mode and operational mode defined. The trunk is not usable yet. You must add circuits to the trunk.

3. Add circuits using the **portcfg fcipcircuit** command. The command requires the source and destination IP addresses that you assigned to ports in step 1, as well as bandwidth assignments. The following example commands create six circuits for the FCIP trunk that you created in step 2. Each circuit provides a fixed 1000 Mbps (1 Gigabit) maximum usable bandwidth.

   Site A

```
Site A:
portcfg fcipcircuit 16 create 0 --remote-ip 192.168.0.64 --local-ip 192.168.0.63 -b 1000000 -B
1000000
portcfg fcipcircuit 16 create 1 --remote-ip 192.168.1.64 --local-ip 192.168.1.63 -b 1000000 -B 1000000
portcfg fcipcircuit 16 create 2 --remote-ip 192.168.2.64 --local-ip 192.168.2.63 -b 1000000 -B 1000000
portcfg fcipcircuit 16 create 3 --remote-ip 192.168.3.64 --local-ip 192.168.3.63 -b 1000000 -B 1000000
portcfg fcipcircuit 16 create 4 --remote-ip 192.168.4.64 --local-ip 192.168.4.63 -b 1000000 -B 1000000
portcfg fcipcircuit 16 create 5 --remote-ip 192.168.5.64 --local-ip 192.168.5.63 -b 1000000 -B 1000000
```

   Site B

```
portcfg fcipcircuit 16 create 0 --remote-ip 192.168.0.63 --local-ip 192.168.0.64 -b 1000000 -B 1000000
portcfg fcipcircuit 16 create 1 --remote-ip 192.168.1.63 --local-ip 192.168.1.64 -b 1000000 -B 1000000
portcfg fcipcircuit 16 create 2 --remote-ip 192.168.2.63 --local-ip 192.168.2.64 -b 1000000 -B 1000000
portcfg fcipcircuit 16 create 3 --remote-ip 192.168.3.63 --local-ip 192.168.3.64 -b 1000000 -B 1000000
portcfg fcipcircuit 16 create 4 --remote-ip 192.168.4.63 --local-ip 192.168.4.64 -b 1000000 -B 1000000
portcfg fcipcircuit 16 create 5 --remote-ip 192.168.5.63 --local-ip 192.168.5.64 -b 1000000 -B 1000000
```

4. To display the results of creating the tunnel and circuits from the preceding steps, use the **portshow fciptunnel all -c** command. Tunnel status for Site A and Site B displays the same.

```
switch63:root> portshow fciptunnel all -c
-------------------------------------------------------------------------------
 Tunnel Circuit  OpStatus  Flags    Uptime  TxMBps  RxMBps ConnCnt CommRt  Met
-------------------------------------------------------------------------------
 16     -         Up      c------    4m22s    0.00    0.00    1      -      -
 16     0 ge0     Up      ---4--s    4m22s    0.00    0.00    1  1000/1000  0
 16     1 ge1     Up      ---4--s    4m12s    0.00    0.00    1  1000/1000  0
 16     2 ge2     Up      ---4--s     4m2s    0.00    0.00    1  1000/1000  0
 16     3 ge3     Up      ---4--s    3m50s    0.00    0.00    1  1000/1000  0
 16     4 ge4     Up      ---4--s    3m34s    0.00    0.00    1  1000/1000  0
 16     5 ge5     Up      ---4--s    2m10s    0.00    0.00    1  1000/1000  0
-------------------------------------------------------------------------------
 Flags:  tunnel: c=compression m=moderate compression a=aggressive compression
                 A=Auto compression f=fastwrite t=Tapepipelining F=FICON
                 T=TPerf i=IPSec l=IPSec Legacy
 Flags: circuit: s=sack v=VLAN Tagged x=crossport 4=IPv4 6=IPv6
                 L=Listener I=Initiator
```

Modifying an FCIP tunnel

FCIP tunnel characteristics and options can be modified as needed using the **portCfg fcipTunnel** command with the **modify** option. The command syntax is as follows:

portCfg fciptunnel *ve_port* **modify** *options*

The VE_Port variable indicates the specific VE_Port to which each tunnel is assigned. The VE_Port number serves as the tunnel ID. The range is 16 through 23 for a Brocade 7800 switch and 12 through 31 for the FX8-24 blade.

The options variable indicates the choice of options listed and described in Creating an FCIP tunnel on page 68

NOTE
When you use **portcfg fciptunnel** to modify the circuit options, the changes apply only to circuit 0.

CAUTION

Using the modify option may disrupt traffic on the specified FCIP tunnel for a brief period of time.

Modifying an FCIP circuit

FCIP circuit characteristics and options can be modified as needed using the **portCfg fcipcircuit** command with the modify option. The general command syntax is as follows:

portCfg fcipcircuit *ve_port* modify *circuit_id options*

ve_port	Each FCIP tunnel or trunk is assigned to a specific VE_Port. The VE_Port number serves as the tunnel ID. Specify the VE_Port of the tunnel that contains the FCIP circuit you want to modify. The range for VE_Ports varies for extension switches and blades. Refer to the section for your switch or blade in Tunnel and circuit requirements on page 47 for information.
circuit_id	The numeric ID assigned when the circuit was created.
options	Options are as listed and described in Creating an FCIP tunnel on page 68

NOTE
You can modify all circuits, including circuit 0, using the **portCfg fcipcircuit** command.

For full details on syntax and using this command, refer to the *Fabric OS Command Reference*.

Deleting an IP interface

You can delete an IP interface using the **portcfg ipif** command with the **delete** option. The command syntax is as follows:

portcfg ipif *[slot/]ge n* delete *ipaddr*

For full details on syntax and using this command, refer to the *Fabric OS Command Reference*.

NOTE
You cannot delete an IP interface if there is a tunnel or circuit configured to use it. Be sure to delete all tunnels, circuits, and IP routes using an interface before deleting it.

Deleting an IP route

You can delete an IP route to a gateway destination IP address using the **portcfg iproute** command with the delete option. The command syntax is as follows for both IPv4 and IPv6 addressing:

portcfg iproute *[slot/]ge n* delete *dest_ipv4* netmask *mask*

portcfg iproute *[slot/]ge n* delete *dest_ipv6/prefix_len*

For full details on syntax and using this command, refer to the *Fabric OS Command Reference*.

NOTE
You cannot delete an IP route if there is a tunnel, circuit, or IP interface configured to use it. Be sure to delete all tunnels, circuits, and IP interface using an IP route before deleting the IP route.

Deleting an FCIP trunk

When you delete an FCIP trunk, you also delete all associated FCIP circuits. Use the **portCfg fciptunnel** command with the delete option to delete FCIP tunnels. The command syntax is as follows:

portcfg fciptunnel *ve_port* delete

For full details on syntax and using this command, refer to the *Fabric OS Command Reference*.

CAUTION

The fciptunnel delete command does not prompt you to verify your deletion. Be sure you want to delete the tunnel before you press Enter.

NOTE
You must delete an FCIP tunnel before you can delete an IP route that it uses and the IP interface that uses the route.

Deleting an FCIP circuit

You can delete individual FCIP circuits using the **portCfg fcipcircuit** command with the delete option. The command syntax is as follows:

portcfg fcipcircuit *ve_port* delete *circuit_id*

For full details on syntax and using this command, refer to the *Fabric OS Command Reference*.

Configuring PP-TCP-QoS priorities over an FCIP trunk

Per-Priority TCP QoS (PP-TCP-QoS) prioritizes FC traffic flows between initiators and targets within an FCIP tunnel to optimize bandwidth and performance.

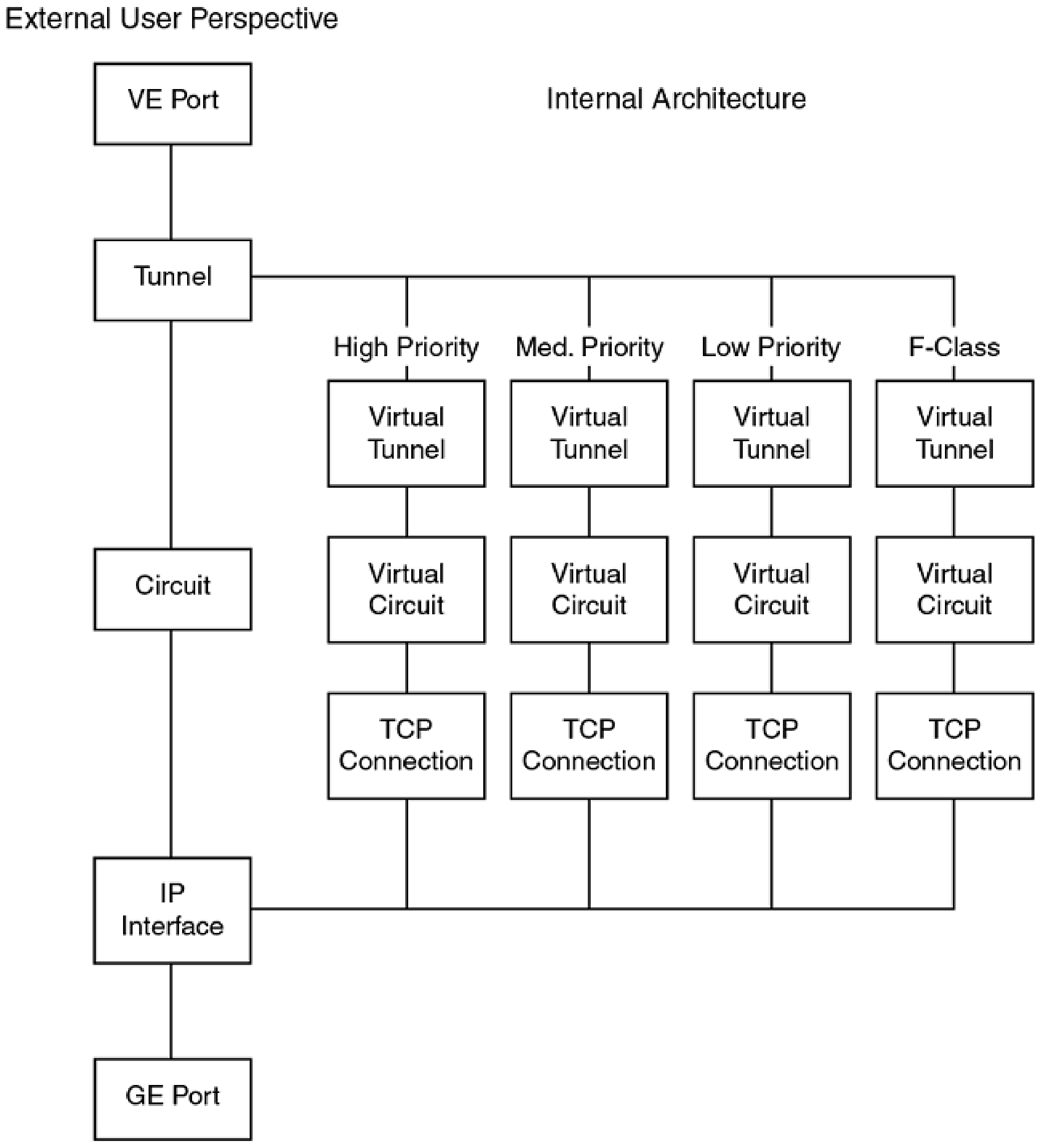
Each circuit has four TCP connections that manage traffic over an FCIP tunnel, as illustrated in the figure below. Each circuit handles one of the following priority traffic types.

- F class - F class is the highest priority, and is assigned bandwidth as needed at the expense of lower priorities, if necessary. This is referred to as strict priority.
- QoS high - The default priority value is 50 percent of the available bandwidth.
- QoS medium - The default value is 30 percent of the available bandwidth.
- QoS low - The default value is 20 percent of the available bandwidth.

QoS high, medium, and low priority traffic are assigned a percentage of available bandwidth based on priority level. QoS priority is based on the Virtual Circuit (VC) that carries data into the FCIP DP complex. For example, if data enters on a high VC, it is placed on a high TCP connection; if it enters on a low VC, then it is placed on the low TCP circuit. Data is assigned to the proper VC based on zone name prefix.

The following figure illustrates the internal architecture of TCP connections that handle PP-TCP-QoS. Note that this illustrates a tunnel containing a single circuit only.

FIGURE 18 TCP connections for handling QoS



Modifying default priority values

You can modify the default QoS priority values on Brocade extension switches and blades. Note that this only changes the QoS priority distribution in the tunnel and does not reconfigure the fabric.

Change the priority percentages on 8 Gbps extension platforms using the optional percentage tunnel argument for the **portcfg fciptunnel** create and **portcfg fciptunnel** modify commands. When configuring QoS percentages for each level, remember the following:

- The three values must equal 100 percent.
- A minimum of 10 percent is required for each level.
- QoS priority settings must be the same on each end of the tunnel.

NOTE
Priorities are enforced only when there is congestion on the network. If there is no congestion, all traffic is handled at the same priority.

The following command sets the QoS priority ratios on VE_Port 12 to high (50%), medium (40%) and low (10%) priorities respectively.

```
portcfg fciptunnel 1/12 create --qos-bw-ratio 50,40,10
```

The following command displays details of the FCIP tunnel configuration, including set QoS percentages.

```
portshow fciptunnel 1/12
```

For more information on using Fabric OS commands, optional arguments, and command output, refer to the *Fabric OS Command Reference*.

Using FCIP with logical switches

Configuring FCIP tunnels and other components in switches enabled for Virtual Fabrics is somewhat different than on switches not enabled for Virtual Fabrics. This section provides a brief overview of common logical switch concepts and terminology followed by the specifics of configuring FCIP on logical switches.

Logical switch overview

The logical switch feature allows you to divide a physical chassis into multiple fabric elements. Each of these fabric elements is referred to as a logical switch. Each logical switch functions as an independent self-contained FC switch. Each chassis can have multiple logical switches.

Logical switches are used to take advantage of multiple VE_Ports. Emulation with ECMP (multiple same-cost VE_Ports) to the same domain are not supported in all emulation modes.

Default logical switch

Virtual Fabrics allows Ethernet ports in the default switch to be shared among VE_Ports in any logical switch. To use the Virtual Fabrics features, you must first enable Virtual Fabrics on the switch. Enabling Virtual Fabrics creates a single logical switch in the physical chassis. This logical switch is called the default logical switch, and it initially contains all of the ports in the physical chassis. After you enable Virtual Fabrics, you can create additional logical switches. The number of logical switches that you can create depends on the switch model.

After you create logical switches, the chassis appears as multiple independent logical switches. All of the ports continue to belong to the default logical switch until you explicitly move them to other logical switches. The default logical switch always exists. You can add and delete other logical switches, but you cannot delete the default logical switch unless you disable Virtual Fabrics.

Creating logical switches

To create logical switches and logical fabrics, you must perform the following steps.

1. Enable Virtual Fabrics mode on the switch using instructions in the "Managing Virtual Fabrics" chapter of the *Fabric OS Administrator's Guide*.
2. Configure logical switches to use basic configuration values using instructions in the "Managing Virtual Fabrics" chapter of the *Fabric OS Administrator's Guide*.
3. Create logical switches using instructions for creating a logical switch or base switch in the "Managing Virtual Fabrics" chapter of the *Fabric OS Administrator's Guide*.

Port assignment

Initially, all ports belong to the default logical switch. When you create additional logical switches, they are empty and you can assign ports to those logical switches. As you assign ports to a logical switch, the ports are moved from the default logical switch to the newly created logical switch. Following are some requirements for assigning ports:

- A given port can be in only one logical switch.
- You can move ports from one logical switch to another.
- A logical switch can have as many ports as are available in the chassis.
- Ports with defined configuration settings in a logical switch or the default switch cannot be moved to another logical switch without first deleting the current settings. For example, you cannot move a VE_ Port that with a defined FCIP tunnel in the default switch or a logical switch to a different logical switch until you delete the FCIP circuits and the FCIP tunnel in the logical switch currently containing the port that you want to move. Similarly, you cannot move a GE_Port between logical switches until all IP routes and IP interfaces have been deleted in the logical switch currently containing the port that you want to move.

Use the **lsCfg** --config *slot/ge _port* command to move ports from one logical switch to a different logical switch. The FID is the fabric ID of the logical switch where you want to move the ports. The ports are automatically removed from the logical switch where they are currently assigned.

As a recommended best practice for FCIP, leave Ethernet interfaces in the default logical switch and do not move them to another logical switch. There is no reason to move them because of the Ethernet Port Sharing (EPS) feature for FCIP. A VE_Port in any logical switch context can use an Ethernet interface in the default switch. In addition, by moving a physical port from the default switch to a logical switch, it will not be available to tunnels configured in other logical switches. Refer to Ethernet Port sharing on page 95 for details.

Logical switches and fabric IDs

When you create a logical switch, you must assign it a fabric ID (FID). The fabric ID uniquely identifies the logical switch within a chassis and indicates the fabric to which the logical switch belongs. You cannot define multiple logical switches with the same fabric ID within the chassis. A logical switch in one chassis can communicate with a logical switch in another chassis (or to a switch not enabled for logical switches) only if the switches have the same fabric ID (FID). The default logical switch is initially assigned FID 128, which can be changed.

Only logical switches with the same FID can form a logical fabric. If you connect two logical switches with different FIDs, the link between the switches segments.

Create logical switches using the **lsCfg** command. For details, refer to the instructions for creating a logical switch or base switch section in the *Fabric OS Administrator's Guide* and to the **lsCfg** command in the *Fabric OS Command Reference*.

Logical switch contexts

You can configure features or perform other tasks on a specific logical switch as you would any Fibre Channel switch by entering commands while in the "context" of that logical switch, which is the FID of the logical switch. Note that "128" is sometimes referred to the context for the default switch as that is the initial FID of the default switch when you enable Virtual Fabrics. However, this FID may be changed.

There are two methods for changing to the context of a specific logical switch so that you can perform configuration or other tasks:

- Use the **setcontext** *fabricID* command. This changes the context to a specific logical switch and changes the command line prompt to reflect the new FID. Any commands entered at this prompt are initiated on the default switch with that FID.
- Use **fosexec** --fid *fabricID* -cmd *command* to initiate a specific command on a specific logical switch, where *command* is the command string.

Connecting logical switches

A logical fabric is a fabric that contains at least one logical switch. You can connect logical switches to non-Virtual Fabrics switches and to other logical switches using two methods:

- Through ISLs. For FCIP, the ISL connection is through a tunnel.
- Through base switches and extended ISLs (XISLs). This is supported by the FX8-24 blade only. Refer to Enabling XISL for VE_Ports (FX8-24 blade) on page 101.

For more information on virtual fabrics

For more detail on managing and configuring virtual fabrics, refer to chapter on managing Virtual Fabrics in the *Fabric OS Administrator's Guide*.

FCIP considerations for logical switches

Before creating IPIFs, IP routes, tunnels, and circuits, follow procedures for creating logical switches as outlined in Logical switch overview on page 93 and as detailed in the chapter on managing virual fabrics in the *Fabric OS Administrator's Guide*. Use the following information and instructions for creating FCIP tunnels and other components on logical switches.

Ethernet port sharing

In Fabric OS v 7.0 and later, VE_Ports in different logical switches can share a single Ethernet port (1 GbE, 10 GbE, or 40 GbE) located on the default switch. As a best practice, leave Ethernet interfaces in the default switch even if you will only use a single virtual fabric logical switch. If new VF logical switches are added and need to use the Ethernet interface, then the Ethernet interface doesn't have to be moved back to the default switch.

NOTE
For Fabric OS versions prior to Fabric OS v7.0, in order to use a Ethernet port for an FCIP tunnel, that port must be in the same logical switch as the tunnel's VE_Port.

With Ethernet port sharing, you can have the following configuration, as an example:

- Default switch has port GbE0
- Logical switch 1 has VE17, which has a circuit over GbE0
- Logical switch 2 has VE18, which also has a circuit over GbE0

All of the committed-rate restrictions and bandwidth sharing of the Ethernet ports for ARL remain the same for shared ports in the logical switches. VE_Ports created from shared Ethernet ports initiate as regular VE ISLs in their respective logical switches. VE_Ports do not need to be associated with a specific Ethernet port.

When IPIFs are created for physical ports (including crossports) located in the default switch, these IP interfaces can be used by circuits assigned to tunnels created in other logical switches. This means that multiple VE_Ports in multiple logical switches can use the same Ethernet connection. Although multiple circuits can use the same Ethernet connection, these circuits can be differentiated in the IP network using VLAN tags or access control lists (ACLs) set for the source and destination IP addresses in the circuit. Refer to Managing QoS, DSCP, and VLANs on page 102 for more information on using VLAN tagging for FCIP.

Limitations of Ethernet port sharing

Note the following limitations of port sharing:

- Only Ethernet ports in the default switch can be shared by VE_Ports in different logical switches. A Ethernet port in a non-default switch can only be used by VE_Ports in that same logical switch.
- The GbE ports in other logical switches or ports on the base switch cannot be shared by ports in different logical switches.
- Tunnels created on 7800 switches and FX8-24 blades with a mix of dedicated ports (ports within the same logical switch) and shared ports (ports in the default switch) are not supported.
- When using shared Ethernet interfaces between the default switch and other logical switches, if the default switch is disabled, the Ethernet ports in the default switch will also be disabled. This will impact all tunnels in the other logical switches using the Ethernet interfaces.

Port sharing example

This section illustrates an example of port sharing on an FX8-24 blade. The following output for the **portshow ipif all** command illustrates IP interfaces, IP routes, and crossports configured for ports in the default logical switch and tunnels and circuits on two different logical switches that use these configurations.

Note the following about the configuration detailed in the output:

- This example is for FCIP configuration on a FX8-24 blade.
- There are three logical switches:
  - LS 0 has FID 128 and is the default switch.
  - LS 2 has FID 50.
  - LS 4 has FID 70.
- IP interfaces and IP routes for these IPIFs were created for xge0 and xge1. The **portcfg - -ipif** and **portcfg - - iproute** commands were issued in the default logical switch context where the ports reside. Refer to Configuring IPIFs and IP routes on page 97 for more information.
- Crossports were configured for both xge0 and xge1 on the default switch. Refer to Crossports on page 33 for more information.
- A tunnel with VE_Port 22 and circuits was created on LS 2. VE_Port 22 was first moved to LS 2, and the **portcfg fciptunnel** commands to configure the tunnel and circuits were issued in the context for LS 2 (FID 50). Refer to Moving ports between logical switches on page 99 and Configuring tunnels and circuits on page 98 for more information.
- A tunnel with VE_Port 12 and circuits was created on LS 4. VE_Port 12 was first moved to LS 4, and the **portcfg fciptunnel** commands to configure the tunnel and circuits were issued in the context for LS 4 (FID 70). Refer to Moving ports between logical switches on page 99 and Configuring tunnels and circuits on page 98 for more information.

```
CURRENT CONTEXT -- LS: 0, FID: 128 *NOTE this
is the default switch.*
portshow ipif all      :
Port: 1/xge0
Interface IP Address       /Pfx      MTU  VLAN  Flags
```

```
--------------------------------------------------------------
    0     10.255.4.80     /24       1500   0   U R M
    1     10.255.4.81     /24       1500   0   U R M
    2     10.255.4.90     /24       1500   0   U R M (crossport)
    3     10.255.4.94     /24       1500   0   U R M (crossport)
Port: 1/xge1
  Port    IP Address      /Pfx      MTU   VLAN  Flags
--------------------------------------------------------------
    0     10.255.4.111    /24       1500    0   U R M
    1     10.255.4.119    /24       1500    0   U R M
    2     10.255.4.70     /24       1500    0   U R M (crossport)
    3     10.255.3.90     /24       1500    0   U R M (crossport)

portshow iproute all      :
Port:
Port      IP Address      /Pfx   Gateway        Flags
-------------------------------------------------------------
1/xge0    10.164.88.119   /24    10.255.4.21    0 U G S
1/xge0    10.164.88.112   /24    10.255.4.79    0 U G S
1/xge0    10.164.88.130   /24    10.255.4.100   0 U G S (crossport)
1/xge0    10.164.88.136   /24    10.255.4.104   0 U G S (crossport)
1/xge     10.164.88.75    /24    10.255.4.110   0 U G S
1/xge     10.164.88.77    /24    10.255.4.115   0 U G S
1/xge     10.164.88.114   /24    10.255.4.65    0 U G S (crossport)
1/xge     10.164.88.115   /24    10.255.4.69    0 U G S (crossport)
CURRENT CONTEXT -- LS: 2, FID: 50 *Note that this is one of the logical
switches (not the default switch).*
portshow fciptunnel all -c:
-------------------------------------------------------------------------------
Tunnel Circuit  OpStatus  Flags    Uptime  TxMBps  RxMBps ConnCnt CommRt  Met
-------------------------------------------------------------------------------
1/22   -         Up      cft----   14d18h  226.60    2.73    5      -      -
1/22   0 1/xge0  Up      ---4v-s 7d17h34m   64.80    0.78    7  1000/3000  0
1/22   1 1/xge0  Up      ---4v-s  7d5h24m   48.59    0.59    7  1000/2000  0
1/22   2 1/xge1  Up      ---4vxs 7d17h34m   64.60    0.78    7  1000/3000  0
1/22   3 1/xge1  Up      ---4vxs  7d5h24m   48.60    0.58    7  1000/2000  0
-------------------------------------------------------------------------------
Flags (tunnel): M=MainTunnel  L=LocalBackup  R=RemoteBackup
                i=IPSec f=Fastwrite T=TapePipelining F=FICON r=ReservedBW
                A=AdvCompr L=LZCompr d=DeflateCompr  D=AggrDeflateCompr
     (circuit): h=HA-Configured v=VLAN-Tagged p=PMTU 4=IPv4 6=IPv6
                ARL a=Auto r=Reset s=StepDown t=TimedStepDown


CURRENT CONTEXT -- LS: 4, FID: 70 *Note that this is a different logical switch
(and not the default switch).*
portshow fciptunnel all -c     :
-------------------------------------------------------------------------------
Tunnel Circuit  OpStatus  Flags    Uptime  TxMBps  RxMBps ConnCnt CommRt  Met
-------------------------------------------------------------------------------
1/12   -         Up      c--F---   19d15h    0.00    0.00    1      -      -
1/12   0 1/xge0  Up      ---4vxs 7d17h34m    0.00    0.00    3  1000/3000  0
1/12   1 1/xge0  Up      ---4vxs  7d5h24m    0.00    0.00    4  1000/2000  1
1/12   2 1/xge1  Up      ---4v-s 7d17h34m    0.00    0.00    3  1000/3000  0
1/12   3 1/xge1  Up      ---4v-s  7d5h24m    0.00    0.00    4  1000/2000  1
-------------------------------------------------------------------------------
Flags:  tunnel: c=compression m=moderate compression a=aggressive compression
                 A=Auto compression f=fastwrite t=Tapepipelining F=FICON
                 T=TPerf i=IPSec l=IPSec Legacy
Flags: circuit: s=sack v=VLAN Tagged x=crossport 4=IPv4 6=IPv6
                 L=Listener I=Initiator
```

Configuring IP interfaces and IP routes

The following are example configures IP interfaces (ipif) and IP routes (iproutes) for ports that reside on the default switch and creating tunnels and circuits on a different logical switch that use these IP interfaces.

You must issue the **portcfg ipif** and **portcfg iproute** commands in the logical switch context where the Ethernet port resides. If the Ethernet port is in the default switch, then the commands must be entered from the default switch context. If the Ethernet ports are in a logical switch other than the default switch, you must issue the commands in that context. In the latter case, the Ethernet ports cannot be used by tunnels created in any other logical switch in the chassis.

In the following example, port ge0 on an FX8-24 blade in slot 8 of a DCX-4S is on the default switch. The default switch FID in this case is 128.

1. If you are in a different logical switch context than the default switch, set the context to 128 using the **setcontext 128** command.

   ```
   sw0:FID60:admin>setcontext 128
   ```

2. Enter the **portcfg ipif** command to create the interface on the port.

   ```
   sw0:FID128:admin>portcfg ipif 8/ge0 create 192.168.1.24 netmask 255.255.255.0 mtu
   1500
   ```

3. Configure an IP route if necessary using the **portcfg iproute** command in the FID 128 context.

   The following command creates an IP route to destination network 192.168.11.0 for port ge0 on the FX8-24 blade in slot 8. The route is through local gateway 192.168.1.1.

   ```
   sw0:FID128:admin> portcfg iproute 8/ge0 create 192.168.11.0 netmask 255.255.255.0
   192.168.1.1
   ```

Other than issuing commands for IP interfaces and IP routes from the correct logical switch context, other aspects of the commands used in this procedure are the same as for any switch. For more information, refer to Configuring an IPIF on page 65 and Configuring an IP route on page 66.

Configuring tunnels and circuits

To configure a tunnel on a logical switch other than the default switch, you must first move the VE_Port to the logical switch from the default switch, and then create the tunnel and circuits in that logical switch context. Issue the **portcfg fciptunnel** command in the context of the logical switch where the VE_Port resides. In the following example, the VE_Port resides on the default switch with FID 128. A tunnel has not been configured yet using this VE_Port.

Following are example steps to configure a tunnel from a Brocade FX8-24 blade to a Brocade 7800 switch. Other than issuing commands to move VE_Ports and to create tunnels and circuits from the correct logical switch context, other aspects of configuring tunnels and circuits are the same for any switch. For more information, refer to Creating an FCIP tunnel on page 68 and Creating additional FCIP circuits on page 82.

1. Move the VE_Port from the default switch to the logical switch with FID 60. VE_Port 12 is available through physical port ge0 located on the default switch.

   ```
   sw0:FID128:admin>lscfg --config 60 8/12
   ```

2. Set the context to the logical switch with FID 60 using the following command.

   ```
   sw0:FID128:admin&gt;setcontext 60
   ```

3. Create a tunnel endpoint on the new logical switch for the FX8-24 blade using the IP interface created for port ge0 on the default logical switch for the blade and a destination address for a remote Brocade 7800 switch. In the following example, the destination address (192.168.11.78) is specified first, followed by the source address (192.168.1.24). ARL minimum (-b) and maximum (-B) committed rates are specified for circuit 0, which is the default circuit created automatically when you configure a tunnel.

   ```
   sw0:FID60:admin> portcfg fciptunnel 8/12 create --remote-ip 192.168.11.78 --local-
   ip 192.168.1.24 -b 5500 -B 6200
   ```

4. Create a tunnel endpoint on the Brocade 7800 switch using the **portcfg fciptunnel** command. Note that the Brocade 7800 switch is not enabled for Virtual Fabrics.

   ```
   switch:admin> portcfg fciptunnel 16 create --remote-ip 192.168.1.24 --local-ip
   192.168.11.78 -b 5500 -B 6200
   ```

5. Create an additional circuit for the FX8-24 end of the tunnel using the following command.

   ```
   sw0:FID60:admin> portcfg fcipcircuit 8/12 create 1 --remote-ip 192.168.11.78 --
   local-ip 192.168.1.25 -b 15500 -B 62000
   ```

6. Create the circuit on the Brocade 7800 end of the tunnel using the following command.

   ```
   switch:admin> portcfg fcipcircuit 16 create 1 --remote-ip 192.168.1.25 --local-ip
   192.168.11.78 -b 15500 -B 62000
   ```

Moving ports between logical switches

To move ports between logical switches, use the following command:

lsCfg --config *FID slot/port*

- The *FID* variable is the Fabric ID of the logical where port is moving to.
- The *slot* number is required for the Brocade FX8-24 blade. Omitted on the Brocade 7800 and the Brocade 7840 switch.
- The *port* number is the FC, VE, or GE port number. For the Brocade 7800 switch, GbE ports are ge0 through ge5. For the Brocade FX8-24 blade, XGE (10 GbE) ports are xge0 or xge1 and GbE ports are ge0-ge9. For the Brocade 7840 switch, 40 GbE ports are ge0-1 and 10 GbE ports are ge2 through ge17.

The following are considerations for moving ports between logical switches when using FCIP:

- As a recommended best practice for FCIP, leave Ethernet interfaces in the default logical switch and do not move them to another logical switch. There is no reason to move them because of the Ethernet Port Sharing feature for FCIP. A VE_Port in any logical switch context can use an Ethernet interface (GbE or XGE port) in the default switch. In addition, by moving a physical port from the default switch to a logical switch, it will not be available to tunnels configured in other logical switches.
- The 1 GbE ports (Brocade 7800 switch and FX8-24 blade), 10 GbE ports (FX8-24 blade and Brocade 7840 switch), 40 GbE ports (Brocade 7840 switch only), and VE_Ports can be part of any logical switch. They can be moved between any two logical switches unless they are members of a circuit configuration.
- Because Ethernet ports and VE_Ports are independent of each other, both must be moved in independent steps. You must delete the configuration on VE_Ports and Ethernet ports before moving them between logical switches.
- You must move a VE_Port from the logical switch where it resides to a new logical switch in order to create an FCIP tunnel for the new logical switch. VE_Ports can be moved to any logical switch independent of the location of the physical Ethernet port.

Displaying logical switch configurations

You can display the logical switch configuration for a switch and the Ethernet ports located in each logical switch using the **lscfg** --show -ge command. The following output shows that all Ethernet ports are located in the default switch (FID 128).

```
DCX68:FID128:root> lscfg --show -ge
Created switches:  128(ds)  10  60  68  127
Slot      1     2     3     4     5     6     7     8     9   10   11   12
--------------------------------------------------------------------------
Port
0    | 128 |     |     |     |     |     |     |     |     |   |   | 128 |
1    | 128 |     |     |     |     |     |     |     |     |   |   | 128 |
2    | 128 |     |     |     |     |     |     |     |     |   |   | 128 |
3    | 128 |     |     |     |     |     |     |     |     |   |   | 128 |
4    | 128 |     |     |     |     |     |     |     |     |   |   | 128 |
5    | 128 |     |     |     |     |     |     |     |     |   |   | 128 |
6    | 128 |     |     |     |     |     |     |     |     |   |   | 128 |
7    | 128 |     |     |     |     |     |     |     |     |   |   | 128 |
8    | 128 |     |     |     |     |     |     |     |     |   |   | 128 |
9    | 128 |     |     |     |     |     |     |     |     |   |   | 128 |
10   | 128 |     |     |     |     |     |     |     |     |   |   | 128 |
11   | 128 |     |     |     |     |     |     |     |     |   |   | 128 |
```

You can display the logical switch configuration and the VE_Ports assigned to each logical switch using the **lscfg** --show command. The following output shows that besides the default switch with FID 128, other default switches have been created with FID 10, 60, 68, and 127.

Note that some of the VE_Ports for the FX8-24 blade in slot 1 have been moved from the default switch to other logical switches.

```
DCX68:FID128:root> lscfg --show
Created switches:  128(ds)  10  60  68  127
Slot      1     2     3     4     5     6     7     8     9   10   11   12
--------------------------------------------------------------------------
Port
0    | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
1    | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
2    | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
3    | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
4    | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
5    | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
6    | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
7    | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
8    | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
9    | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
10   | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
11   | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
12   |  60 |     |     |     | 128 |     |     | 128 |  60 |   |   |  60 |
13   | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
14   |  10 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
15   | 128 |     |     |     | 128 |     |     | 128 |  60 |   |   | 128 |
16   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
17   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
18   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
19   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
20   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
21   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
22   |  10 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
23   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
24   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
25   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
26   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
27   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
28   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
29   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
30   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
31   | 128 |     |     |     | 128 |     |     | 128 |  10 |   |   | 128 |
```

Brocade 7800 switch considerations and limitations

The following are considerations and limitations for Brocade 7800 switches configured to support Virtual Fabrics:

- Although you can create up to four logical switches on a Brocade 7800 switch, a base switch cannot be created. Therefore, you cannot use the logical switches for XISLs.
- Up to four logical switches will support FICON CUP; however, refer to your system qualification letter-specific limits.
- FCR is not supported on a Brocade 7800 switch enabled with logical switches because the Brocade 7800 has no base switch to support EX_Ports.
- A Brocade 7800 switch configured with multiple logical switches cannot be downgraded to a prior release without deleting all of the non-default logical switches and configurations.

Brocade FX8-24 blade considerations and limitations

The following are considerations and limitations of FX8-24 blades configured to support Virtual Fabrics:

- The number of logical switches that you can create and the limits on logical switch support for FICON CUP depends on the chassis where the FX8-24 is installed. For example, up to eight logical switches can be configured on a blade installed on DCX 8510 platforms. Refer to your chassis specifications for details.

- The total number of VE_Ports in all the logical switches is bound by the maximum VE_Ports on a blade (20) multiplied by the maximum number of blades in a chassis.
- For FX8-24 blade, you can make the logical switch a base switch if you are planning on using an extended interswitch link (XISL) connection between base switches instead of using separate ISL connections from logical switches.

Enabling XISL for VE_Ports (FX8-24 blade)

Another way to connect logical switches is to use extended interswitch links (XISLs) and base switches. When you divide a chassis into logical switches, you can designate one of the switches to be a base switch. A base switch is a special logical switch that is used for interconnecting the physical chassis.

An XISL connection can be created between base switches, instead of using separate ISLs. The base fabric provides the physical connectivity across which logical connectivity will be established. The XISL can carry combined traffic for multiple logical fabrics while maintaining traffic separation for each fabric.

Because of the expense of long-distance links, this feature has particular benefit for the FCIP extension platforms. This feature is supported only on tunnels between Brocade FX8-24 blades running Fabric OS v7.0 and later. The blades can be operating in both 1 Gbps and 10 Gbps modes.

NOTE
Although you can create up to four logical switches on the Brocade 7800 and 7840, you cannot use these for XISLs because base switches cannot be created.

To create a base switch on the Brocade FX8-24 blade, use the *-base* option for the **lsCfg** command when creating a logical switch. To use XISL, refer to instructions for configuring a logical switch to use XISLs in the *Fabric OS Administrator's Guide*.

For the Brocade FX8-24 blade, if an XISL is enabled, it is recommended that you do not configure VE_Ports on both the logical switch and the base switch because FCIP tunnels support only two hops maximum.

Brocade 7840 switch considerations and limitations

Following are considerations and limitations of Brocade 7840 switches configured to support Virtual Fabrics:

- Although you can create up to four logical switches on a Brocade 7840 switch, a base switch cannot be created. Therefore, you cannot use the logical switches for XISLs.
- Up to four logical switches will support FICON CUP; however, refer to your system qualification letter-specific limits.
- FCR is not supported on a Brocade 7840 switch enabled with logical switches because the Brocade 7840 has no base switch to support EX_Ports.
- An FCIP tunnel from a Brocade 7840 switch requires one of the following:
  - A VE_Port and Ethernet port in the same logical switch.
  - A VE_Port in a logical switch and shared Ethernet port in the default switch (best practice).

Managing QoS, DSCP, and VLANs

Quality of Service (QoS) refers to policies for handling differences in data traffic. These policies are based on data characteristics and delivery requirements. For example, ordinary data traffic is tolerant of delays and dropped packets, but real-time voice and video data are not. QoS policies provide a framework for accommodating these differences in data as it passes through a network.

QoS for Fibre Channel traffic is provided through internal QoS priorities. Those priorities can be mapped to TCP/IP network priorities using zone name prefixes and VCs. The different priority TCP sessions can be marked upon egress. The TCP marking is done at the IP layer using Layer 3 Differentiated Services Code Point (DSCP) or at the Ethernet layer within the 802.1Q tag header using 802.1P. There are two options for TCP/IP network-based QoS:

- DSCP
- VLAN tagging and Layer 2 Class of Service (L2CoS)

You can configure QoS, DSCP, and VLAN tagging at the FCIP tunnel and circuit level for data path traffic.

DSCP Quality of Service

Layer 3 Class of Service Differentiated Services Code Point (DSCP) refers to a specific implementation for establishing QoS policies as defined by RFC 2475. DSCP uses six bits of the Type of Service (TOS) field in the IP header to establish up to 64 different values to associate with data traffic priority.

DSCP settings are useful only if IP routers are configured to enforce QoS policies uniformly within the network. IP routers use the DSCP value as an index into a Per Hop Behavior (PHB) table. Control connections and data connections can be configured with different DSCP values. Before configuring DSCP settings, determine if the IP network you are using implements PHB, and consult with the WAN administrator to determine the appropriate DSCP values.

VLANs and Layer 2 Quality of Service

Devices in physical LANs are constrained by LAN boundaries. They are usually in close proximity to each other, and share the same broadcast and multicast domains. Physical LANs often contain devices and applications that have no logical relationship. Also, when logically related devices and applications reside in separate LAN domains, they must be routed from one domain to the other.

A virtual local area network (VLAN) can reside within a single physical network, or it can span several physical networks. Related devices and applications that are separated by physical LAN boundaries can reside in the same VLAN. Also, a large physical network can be broken into smaller VLANs. VLAN traffic is routed using 802.1Q-compliant tags within an Ethernet frame. The tag includes a unique VLAN ID, and Class of Service (CoS) priority bits. The CoS priority scheme (also called Layer 2 Class of Service or L2CoS) uses three Class of Service (CoS or 802.1P) priority bits, allowing eight priorities. Consult with your WAN administrator to determine usage.

Managing DSCP and VLAN support on FCIP circuits

When VLAN tag is created on an FCIP circuit, all traffic over that circuit will use the specified VLAN. VLAN tagging of ingress FCIP traffic is often used to identify multiple circuits of different VE_Ports coming from a 10 GbE port into a 10 GbE port on a switch or router. If the switch or router has its 10 GbE interface configured as a VLAN trunk and each circuit has its own tagging, then the switch or router can direct the traffic accordingly.

Options listed in the following table are available on the **portcfg fciptunnel** command to enable VLAN support on circuit 0 and on the **portcfg fcipcircuit** command for additional circuits. Only traffic flow over the configured circuit will be tagged with the VLAN tagging and the L2CoS and DSCP options.

TABLE 16 VLAN and DSCP options

Options	Description
VLAN -v - -vlan-tagging *vlan_id* 7800 switch and FX8-24 blade only.	For the Brocade 7800 switch and FX8-24 blade, the *vlan_id* parameter sets the VLAN tag value in the header assigning the traffic to that specific VLAN. The VLAN tag is an integer value from 1 through 4094. Consult with your WAN administrator to discuss VLAN implementation. For the Brocade 7840 switch, set the VLAN tag value using the vlan *vlan_id* option in the **portcfg ipif** command. Refer to Configuring an IPIF on page 65 for more information.
L2CoS - - L2cos-f-class *n* - - L2cos-high *n* - - L2cos-medium *n* - - L2cos-low *n*	The IEEE 802.1P specification establishes eight levels of L2CoS priority. A value of 7 is the highest priority, and a value of 0 is the lowest priority and the default. Values can be applied on a per-QoS basis for ingress traffic on each FCIP circuit. Consult with your WAN administrator to discuss L2CoS implementation.
DSCP - -dscp-f-class *n* - -dscp-high *n* - -dscp-medium *n* - -dscp-low *n*	The DSCP options allow you to specify a DSCP marking tag on a per-QoS basis for ingress traffic on each FCIP circuit. On the extension switches and blades, only traffic going over the FCIP tunnel is marked. A decimal value from 0 through 63 may be used to specify the DSCP marking tag. Consult with your WAN administrator to discuss DSCP implementation before assigning a DSCP marking tag.

VLAN tagging examples

The following example shows the VLAN tag option on the **portcfg fciptunnel create** command. The VLAN tag applies only to circuit 0 because a circuit was not identified and this is the default circuit for a tunnel.

```
switch:admin> portcfg fciptunnel 16 create --remote-ip 192.168.2.20 --local-ip
192.168.2.10 -b 100000 -B 150000-v 100
Operation Succeeded
```

The following example creates an additional FCIP circuit with a different VLAN tag.

```
switch:admin> portcfg fcipcircuit 16 create 1 --remote-ip 192.168.2.21 --local-ip
192.168.2.11 -b 100000 -B 150000 -v 200
Operation Succeeded
```

The following example shows the **portcfg fcipcircuit modify** command that changes the VLAN tag and L2CoS levels for circuit 0. Parameters are the same for both the create and modify options.

```
switch:admin> portcfg fcipcircuit 16 modify 0 -v 300 --l2cos-f-class 7 --l2cos-high 5
--l2cos-medium 3 --l2cos-low 1
```

The following example shows the **portcfg fcipcircuit modify** command that changes the DSCP values for circuit 0. Parameters are the same for both the create and modify options.

```
switch:admin> portcfg fcipcircuit 16 modify 0 --dscp-f 32 --dscp-h 16 --dscp-m 8 --
```

```
dscp-l 4
Operation Succeeded
```

The following example shows the use of the **portshow** command to display the tunnel and circuit values. Use the -c option as shown to include circuit values.

```
switch:admin> portshow fciptunnel 16 -c
```

When both DSCP and L2CoS are used

If an FCIP tunnel or circuit is VLAN tagged, both DSCP and L2CoS may be tagged on ingress traffic unless the VLAN is end-to-end with no intermediate hops in the IP network. The following table shows DSCP priorities mapped to L2CoS priorities. This may be helpful when consulting with the network administrator. You can modify DSCP and L2CoS values for different priority traffic when configuring circuits for extension switches and blades.

TABLE 17 Default mapping of DSCP priorities to L2CoS priorities

DSCP priority/bits	L2CoS priority/bits	Assigned to
7 / 000111	1 / 001	Medium QoS
11 / 001011	3 / 011	Medium QoS
15 / 001111	3 / 011	Medium QoS
19 / 010011	3 / 011	Medium QoS
23 / 010111	3 / 011	Medium QoS
27 / 011011	0 / 000	Class 3 Multicast
31 / 011111	0 / 000	Broadcast/Multicast
35 / 100011	0 / 000	Low Qos
39 / 100111	0 / 000	Low Qos
43 / 101011	4 / 100	High QoS
46 / 101110	7 / 111	Class F
47 / 101111	4 / 100	High QoS
51 / 110011	4 / 100	High QoS
55 / 110111	4 / 100	High QoS
59 / 111011	4 / 100	High QoS
63 / 111111	0 / 000	Reserved

Managing the VLAN tag table

The VLAN tag table is used by ingress processing to filter inbound VLAN tagged frames per IP interface. The table is used to determine how to tag a frame that is not already tagged. If a VLAN tagged frame is received from the network and there is no entry in the VLAN tag table for the VLAN ID, the frame is discarded. The per-IP interface VLAN configuration is for non-data path traffic only, such as ICMP or ping commands. If Class F traffic or data path traffic needs to be tagged, it must be done use the following methods:

- For the Brocade 7800 switch and FX8-24 blade, use the -v, - -vlan-tagging options in the **portcfg fcipcircuit create** or **portcfg fcipcircuit modify** commands.
- For the 7840 switch, set the VLAN tag value using the vlan *vlan_id* option in the **portcfg ipif** command.

To tag frames destined for a specific host address, you must create an entry with an exact matching destination address in the table. Only frames destined for that address are tagged with the associated VLAN ID. To tag frames destined for a specific network, you must create a destination address entry for the network. For example, if a destination address of 192.168.100.0 is specified, then all frames destined for the 192.168.100.0 network are tagged with the associated VLAN ID, assuming a network mask of 255.255.255.0. If frames are already VLAN tagged, those tags take precedence over entries in this table.

NOTE
If you do not specify a destination IP address, the destination address defaults to 0.0.0.0, and all frames are tagged with the associated VLAN tag.

1. Connect to the switch and log in using an account assigned to the admin role.
2. For the Brocade 7800 and FX8-24 blade, enter the **portCfg vlantag** command to add or delete entries in the VLAN tag table. The general syntax for the command is as follows:

   portCfg vlantag [add|delete] *ipif_addr vlan_id* L2CoS *dst_IP_addr*

   For full details on syntax and using this command, refer to the *Fabric OS Command Reference*.

   The following example adds an entry that tags all frames from IP address 192.168.10.1 destined for IP address 192.168.20.1 with a VLAN ID of 100, and a L2CoS value of 3.

   ```
   switch:admin> portcfg vlantag 8/ge0 add 192.168.10.1 100 3 192.168.20.1
   ```

   The following example for the FX8-24 blade adds an entry that tags al frames from a crossport with local address 192.168.11.20, VLAN ID of 200, and a LSCoS value of 1.

   ```
   switch:admin> portcfg vlantag 8/xge0 add 192.168.11.20 200 1 -x
   ```
3. For the Brocade 7840 switch, add entries in the VLAN tag table using the vlan *vlan-id* option in the **portcfg ipif** command, which is used to configure an IP interface (IPIF) for circuits that you intend to configure on a Ethernet port. If no VLAN ID is specified for the IP address, no ID is used. The IP address used on the command line will belong to the VLAN Id specified.

   The general syntax for using this option in the command is as follows:

   portcfg ipif *slot/port* .dp0|1 create *src_ipaddr* vlan *vlan_id*

   The following example adds an entry that tags all frames from IP address 192.168.1.10 with a VLAN ID of 100.

   ```
   portcfg ipif ge1.dp0 create 192.168.1.10 vlan 100
   ```

   Note that because GbE ports are shared between DP0 and DP1 on the Brocade 7840 switch, the ge1.dp0 specifies the DP where the command should be sent.

4. To display the VLAN tag configuration, use the **portCfg vlantag** command as follows:

   ```
   switch:admin> portshow vlantag 8/ge0
   switch:admin> portshow vlantag 8/xge0
   ```

   For more details on using the **portCfg vlantag** and **portshow vlantag** commands, refer to the *Fabric OS Command Reference*.

Implementing IPsec over FCIP tunnels

Internet Protocol security (IPsec) uses cryptographic security to ensure private, secure communications over Internet Protocol networks. IPsec supports network-level data integrity, data confidentiality, data origin authentication, and replay protection. It helps secure your SAN against network-based attacks from untrusted computers.

The following sequence of events invokes the IPsec protocol.

1. IPsec and Internet Key Exchange (IKE) policies are created and assigned on peer switches or blades on both ends of the FCIP tunnel.
2. Traffic from an IPsec peer with the lower local IP address initiates the IKE negotiation process.
3. IKE negotiates security association (SA) parameters, setting up matching SAs in the peers. Some of the negotiated SA parameters include encryption and authentication algorithms, Diffie-Hellman key exchange, and SAs.
4. Data is transferred between IPsec peers based on the IPsec parameters and keys stored in the SA database.
5. SA lifetimes terminate through deletion or by timing out. An SA lifetime equates to approximately two billion frames of traffic passed through the SA.

Limitations using IPsec over FCIP tunnels

The following limitations apply to using IPsec:

- Network Address Translation (NAT) is not supported.
- Authentication Header (AH) is not supported.
- IPsec-specific statistics are not supported.
- There is no RAS message support for IPsec.
- IPsec can only be configured on IPv4-based tunnels.
- Older versions of the FX8-24 blade do not support IPsec on VE_Ports 22-31. For these blades, a RASlog warning message will display that blade is not at correct version to support IPsec enabled tunnels on VE_Ports 22 through 31.
- Both ends of the tunnel must use Fabric OS v7.0.0 and later to enable IPsec with Fabric OS v7.0.0 and later.
- IPsec is not allowed with the --connection-type FCIP tunnel option set to anything other than default.

IPsec for the extension switches and blades

Advanced Encryption Standard, Galois/Counter Mode, Encapsulating Security Payload (AES-GCM-ESP) is used as a single, predefined mode of operation for protecting all TCP traffic over an FCIP tunnel. AES-GCM-ESP is described in RFC 4106. The following list contains key features of AES-GCM-ESP:

- Encryption is provided by AES with 256-bit keys.
- The IKEv2 key exchange protocol is used by peer switches and blades for mutual authentication.
- IKEv2 uses UDP port 500 to communicate between the peer switches or blades.
- All IKEv2 traffic is protected using AES-GCM-ESP encryption.
- Authentication requires the generation and configuration of 32-byte pre-shared secrets for each tunnel.
- An SHA-512 hash message authentication code (HMAC) is used to check data integrity and detect third-party tampering.
- Pseudo-random function (PRF) is used to strengthen security. The PRF algorithm generates output that appears to be random data, using the SHA-512 HMAC as the seed value.
- A 2048-bit Diffie-Hellman (DH) group is used for both IKEv2 and IPsec key generation.
- The SA lifetime limits the length of time a key is used. When the SA lifetime expires, a new key is generated, limiting the amount of time an attacker has to decipher a key. Depending on the length of time expired or the length of the data being transferred, parts of a message may be protected by different keys generated as the SA lifetime expires.

  For extension switches and blades, the SA lifetime is approximately eight hours or two billion frames of data. The lifetime is based upon datagrams that have been encrypted over the tunnel regardless of the number of bytes or the time that the tunnel has been up. Once an IPsec SA has been used for 2 billion datagrams, a new SA or re-key sequence is initiated.
- Encapsulating Security Payload (ESP) is used as the transport mode. ESP uses a hash algorithm to calculate and verify an authentication value, and only encrypt the IP payload.
- A circuit in a non-secure tunnel can use the same Ethernet interface as a circuit in a secure tunnel. Each circuit can have a route configured on that Ethernet interface.
- There are no topology restrictions with IPsec enabled.
- Brocade IPsec is a hardware implementation that adds almost no latency to FCIP frame processing.
- Brocade IPsec does not preclude the use of compression or QoS.
- When Brocade IPsec is enabled, it does not degrade FCIP throughput compared to when IPsec is disabled.

Enabling IPsec and IKE policies

IPsec is enabled on the tunnel level, not on the circuit level. For the Brocade 7800 switch and FX8-24 blade, you define and enable IPsec using the --ipsec option of the **portcfg fciptunnel create** and **portcfg fciptunnel modify** commands. The -i option activates IPsec. The -K (preshared-key) option specifies the IKE key. The -l (legacy) option specifies to use the IPsec connection process compatible with Fabric OS releases prior to v7.0.0. Note that the -l option is a disruptive modify request that causes the tunnel to bounce.

On the Brocade 7840 switch, before enabling IPsec on a tunnel, you must first define an IPsec policy using the **portcfg ipsec-policy** command. When defining the IPsec policy, use the -K option to specify the IKE key. Enable the policy on a tunnel using the **--ipsec** *policy* option for the **portcfg fciptunnel** and **portcfg fciptunnel modify** commands. Display the defined IPsec policies, IKE sessions associated with the policy, and other detailed information using the **portshow ipsec-policy** command. Display IPsec configuration on a specific tunnel using the **portshow fciptunnel** command.

The IKE key must be a shared 32-character string. Both ends of the secure tunnel must be configured with the same key string, referred to as a pre-shared key (PSK). If both ends are not configured with the same key, the tunnel will not come up.

The following examples are for the Brocade 7800 switch and FX8-24 blade. They show IPsec and IKE keys enabled for traffic from VE_Ports 16 and 17 across multiple FCIP circuits.

```
portcfg fciptunnel 16 create --remote-ip 192.168.0.90 --local-ip 192.168.0.80 50000 \
-x 0 -d c0 -I -K12345678901234567890123456789012 -l
portcfg fcipcircuit 16 create 1 --remote-ip 192.168.1.90 --local-ip 192.168.1.80 50000 -x 0
portcfg fcipcircuit 16 create 2 --remote-ip 192.168.2.90 --local-ip 192.168.2.80 50000 -x 0
portcfg fcipcircuit 16 create 3 --remote-ip 192.168.3.90 --local-ip 192.168.3.80 50000 -x 0
portcfg fcipcircuit 16 create 4 --remote-ip 192.168.4.90 --local-ip 192.168.4.80 50000 -x 0
portcfg fcipcircuit 16 create 5 --remote-ip 192.168.5.90 --local-ip 192.168.5.80 50000 -x 0
portcfg fciptunnel 17 create --remote-ip 192.168.0.91 --local-ip 192.168.0.81 50000 -x 0 -d \
c0 -I -K12345678901234567890123456789012 -l
portcfg fcipcircuit 17 create 1 --remote-ip 192.168.1.91 --local-ip 192.168.1.81 50000 -x 0
portcfg fcipcircuit 17 create 2 --remote-ip 192.168.2.91 --local-ip 192.168.2.81 50000 -x 0
portcfg fcipcircuit 17 create 3 --remote-ip 192.168.3.91 --local-ip 192.168.3.81 50000 -x 0
portcfg fcipcircuit 17 create 4 --remote-ip 192.168.4.91 --local-ip 192.168.4.81 50000 -x 0
portcfg fcipcircuit 17 create 5 --remote-ip 192.168.5.91 --local-ip 192.168.5.81 50000 -x 0
```

The following command creates an IPsec policy for the Brocade 7840 switch.

```
switch:admin> portcfg ipsec-policy myPolicy1 create  -k
“some test key”
Operation Succeeded.
```

The following command enables the IPsec policy for a Brocade 7840 switch.

```
switch:admin> portcfg tunnel 24 modify --ipsec myPolicy1
Operation Succeeded.
```

The following command displays the policy information on the Brocade 7840 tunnel.

```
switch:admin> portshow fciptunnel -c

Tunnel Circuit OpStatus Flags    Uptime TxMBps  RxMBps ConnCnt CommRt Met
-------------------------------------------------------------------------
 24    -         Up     --i-----  33m4s   0.00    0.00    3      -      -
 24    0 ge2     Up     ---ah--4  33m4s   0.00    0.00    3  1000/1000  0
-------------------------------------------------------------------------
Flags (tunnel): i=IPSec f=Fastwrite T=TapePipelining F=FICON r=ReservedBW
                A=AdvCompr L=LZCompr d=DeflateCompr  D=AggrDeflateCompr
     (circuit): h=HA-Configured v=VLAN-Tagged p=PMTU 4=IPv4 6=IPv6
                ARL a=Auto r=Reset s=StepDown t=TimedStepDown
```

Traffic Isolation Zoning

The Traffic Isolation (TI) Zoning feature allows you to control the flow of inter-switch traffic by creating a dedicated path for traffic flowing from a specific set of source ports (N_Ports). You can use Traffic Isolation Zoning to ensure that requests and responses for FCIP-based applications such as Open Systems Tape Pipelining use the same VE_Port tunnel across a metaSAN.

Traffic isolation is implemented using a special zone, called a Traffic Isolation zone (TI zone). A TI zone indicates the set of N_Ports, E_Ports, and VE_Ports to be used for a specific traffic flow. When a TI zone is activated, the fabric attempts to isolate all inter-switch traffic entering from a member of the zone to only those E_Ports that have been included in the zone. The fabric also attempts to exclude traffic not in the TI zone from using E_Ports or VE_Ports within that TI zone.

For more information and details to configure TI Zoning, refer to the "Traffic Isolation Zoning" chapter in the *Fabric OS Administrator's Guide*.

FCIP Management and Troubleshooting

In-band management

NOTE
In-band management is supported on the Brocade 7800 switch and FX8-24 blade only.

In-band management allows management of an extension switch or blade in conjunction with FCIP traffic through Ethernet ports. This enables a management station located on the WAN side of the FCIP platform to communicate with the control processor (CP) for management tasks, such as SNMP polling, SNMP traps, troubleshooting, and configuration. Through IP forwarding, inband management also allows a management station connected to a LAN through the management port of one extension switch or blade to manage the swiotch or blade at the far end of the network through the WAN.

The in-band management path is achieved by receiving the management traffic from the Ethernet port and transmitting the traffic to the CP through a new interface. The CP then handles the management traffic as it would handle any other management requests from a normal management interface. The in-band management interface is protocol-independent, so any traffic destined for these in-band management interfaces passes through the data processor (DP) to the CP. It is then handled on the CP according to the rules set forth for the normal management interface and follows any security rules that may be in place on the CP.

One in-band management interface can be configured per Ethernet interface to provide redundancy. This allows the management station on the WAN side of the network to have multiple addresses for reaching that switch and provides redundancy if one of the Ethernet ports cannot be reached. Communication is handled through external addresses configured independently for each in-band management interface.

The following functions are not supported by the in-band management interface:

- Downloading firmware
- IPv6 addressing

IP routing

The in-band management interfaces are separate from the existing IP interfaces currently used for FCIP. These interfaces exist on the CP and are added and maintained on the CP routing table to ensure end-to-end connectivity. Because this routing table will be shared among all devices on the CP, including the management interface, precautions must be taken to ensure that proper connectivity is maintained. To ensure proper handling of routes, the in-band management devices should be configured on a different network from the management interface and from every other in-band management interface.

In-band management interface addresses must also be unique and cannot be duplicates of any addresses defined on the Ethernet ports. An in-band management address can exist on the same network as an address defined on one of the GbE ports because the in-band management interfaces use the CP routing table and not the routing table normally used for the GbE ports.

Configuring IP addresses and routes

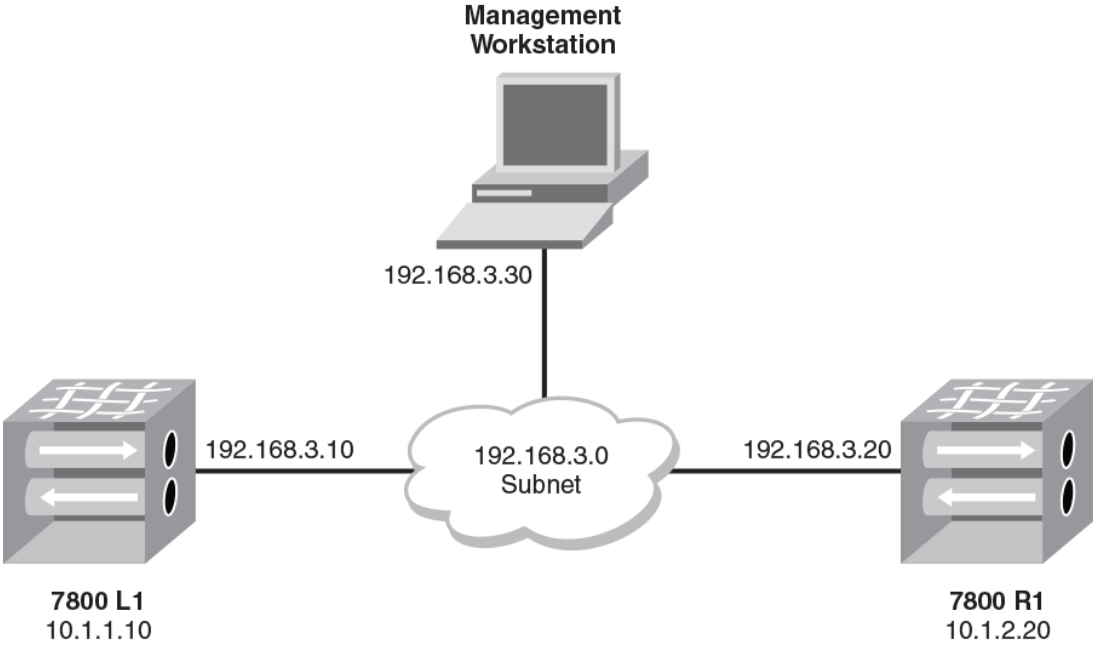
Configure and view IP addresses and routes for in-band management interfaces by using the following Fabric OS commands:

- **portcfg mgmtif** *slot/gePort* **create|delete** *ipAddress netmask mtu*
- **portcfg mgmtif** *slot/gePort* **enable|disable**
- **portshow mgmtif** *slot/gePort*
- **portcfg mgmtroute** *slot/gePort destination netmask gateway*

Management station on the same subnet example

The following figure illustrates an example of configuring in-band management with the management station attached to the same subnet as managed switches. Note that only the IP address is required for each extension switch.

FIGURE 19 Management station configured on the same subnet



7800 LI

Configure the in-band management interfaces.

```
portcfg mgmtif ge0 create 192.168.3.10 255.255.255.0
```

7800 RI

Configure the in-band management interfaces.

```
portcfg mgmtif ge0 create 192.168.3.20 255.255.255.0
```

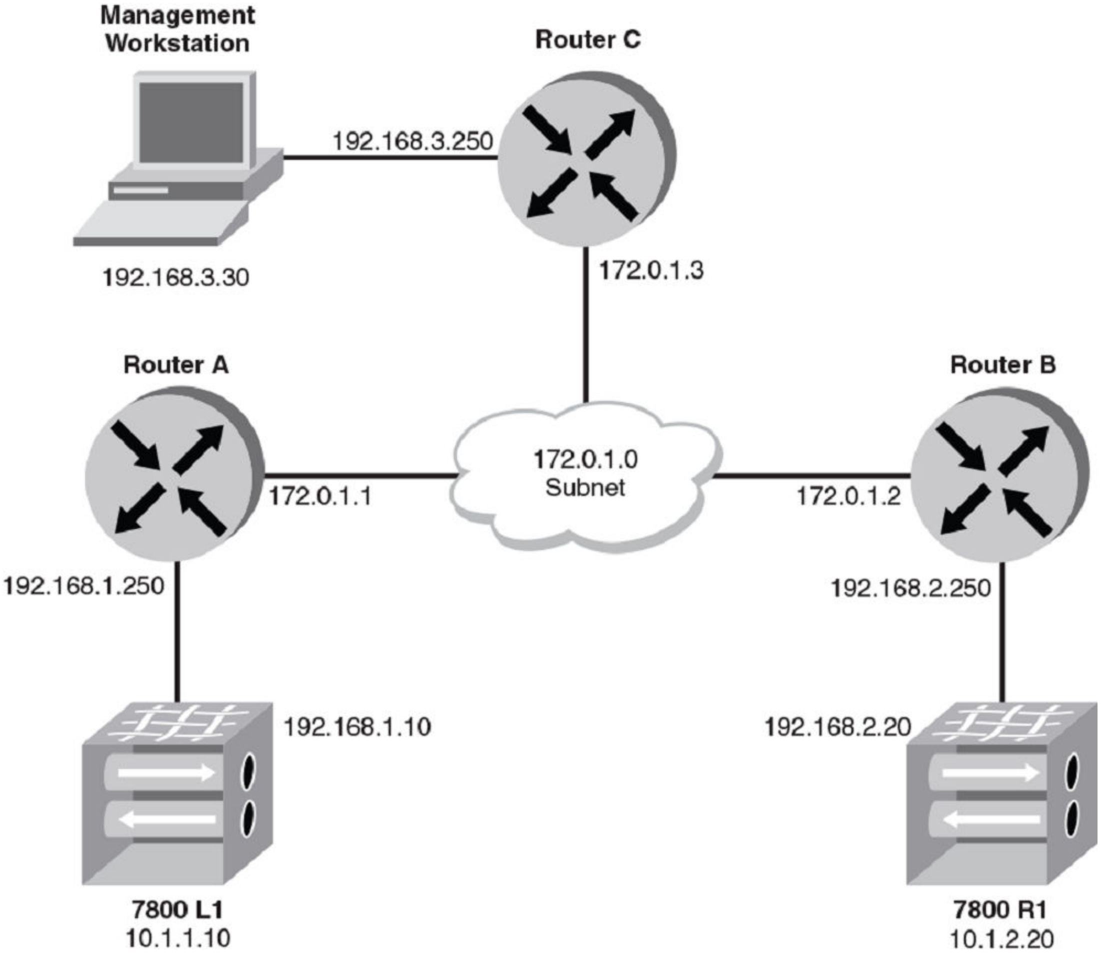
Management Workstation

```
telnet 192.168.3.10
```

Management station on a different subnet example

The following figure illustrates an example configuration consisting of switches and the management station on different networks and attached through a WAN cloud. The routers are assumed to already have route entries to access each other subnet.

FIGURE 20 Management station configured on different subnets



7800 L1

- Configure the in-band management interfaces.

  ```
  portcfg mgmtif ge0 create 192.168.1.10 255.255.255.0
  ```
- Configure the in-band management route for the management station.

  ```
  portcfg mgmtroute ge0 create 192.168.3.0 255.255.255.0 192.168.1.250
  ```

7800 R1

- Configure the in-band management interfaces.

```
portcfg mgmtif ge0 create 192.168.2.20 255.255.255.0
```

- Configure the in-band management route for the management station.

```
portcfg mgmtroute ge0 create 192.168.3.0 255.255.255.0 192.168.2.250
```

Management station

- Add route entries to access the Brocade 7800 external in-band management interfaces.

```
route add 192.168.1.0 netmask 255.255.255.0 gw 192.168.3.250
route add 192.168.2.0 netmask 255.255.255.0 gw 192.168.3.250
```

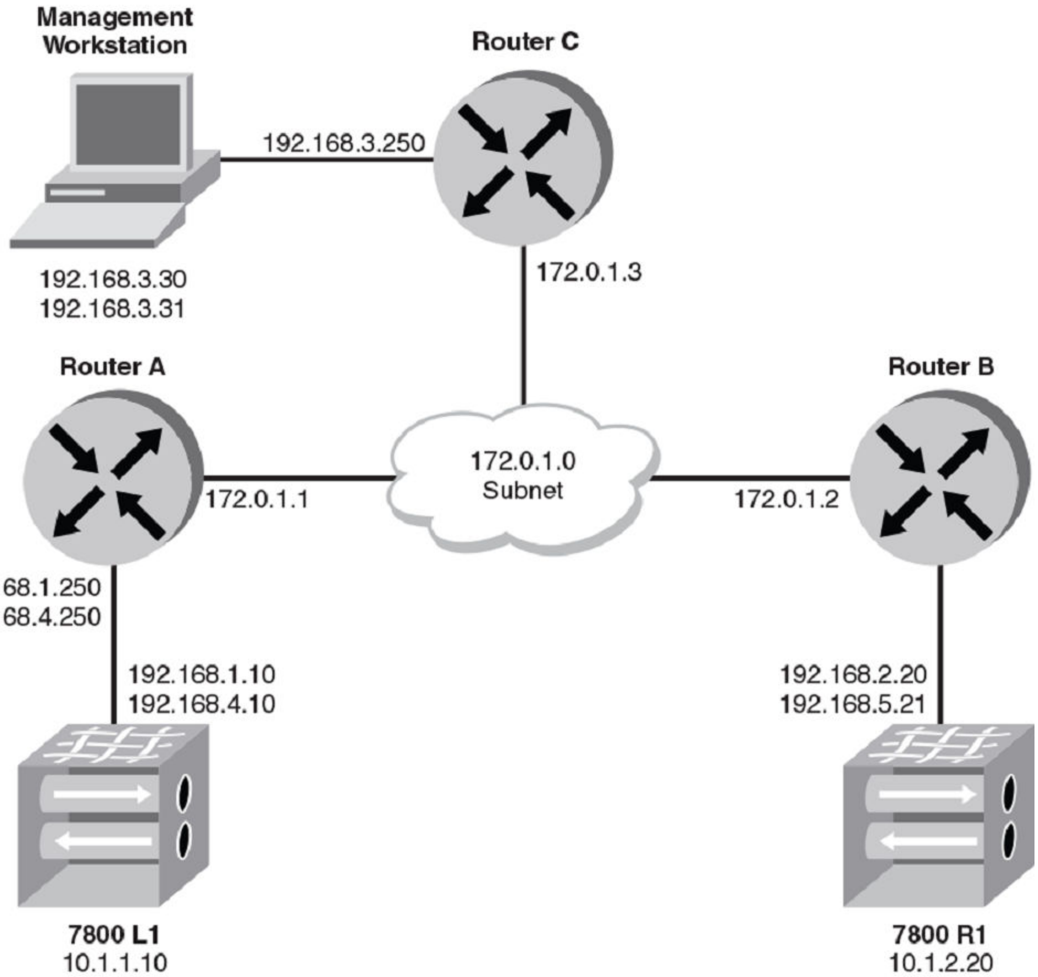
- Access the Brocade 7800 switches through the external in-band management interfaces.

```
telnet 192.168.1.10
```

Redundant connections to the management stations example

In the following figure, because the in-band management interfaces do not support a multi-homing stack, unique addresses must be used on the management station to communicate with different in-band management interfaces. If both management station interfaces are on the same subnet, then host-specific routes must be added on the Brocade 7800 switches.

FIGURE 21 Redundant connections to management station



7800 L1

- Configure the in-band management interfaces.

```
portcfg mgmtif ge0 create 192.168.1.10 255.255.255.0
portcfg mgmtif ge1 create 192.168.4.10 255.255.255.0
```

- Configure the in-band management route for the management workstation.

```
portcfg mgmtif ge0 create 192.168.1.10 255.255.255.0
portcfg mgmtif ge1 create 192.168.4.10 255.255.255.0
```

7800 R1

- Configure the in-band management interfaces.

```
portcfg mgmtif ge0 create 192.168.2.20 255.255.255.0
portcfg mgmtif ge1 create 192.168.5.20 255.255.255.0
```

- Configure the in-band management route for the management workstation.

```
portcfg mgmtroute ge0 create 192.168.3.30 255.255.255.255 192.168.2.250
portcfg mgmtroute ge1 create 192.168.3.31 255.255.255.255 192.168.5.250
```

Management Workstation

- Add route entries to get to the Brocade 7800 external in-band management interfaces.

```
route add 192.168.1.0 netmask 255.255.255.0 gw 192.168.3.250
route add 192.168.2.0 netmask 255.255.255.0 gw 192.168.3.250
route add 192.168.4.0 netmask 255.255.255.0 gw 192.168.3.250
route add 192.168.5.0 netmask 255.255.255.0 gw 192.168.3.250
```

- Access the Brocade 7800 switches through the external in-band management interfaces.

```
telnet 192.168.1.10
```

VLAN tagging support

To add VLAN tag entries to the VLAN tag table for in-band management interfaces, use the --mgmt or -m option with the **portcfg vlantag** command. Complete the following steps.

1. Configure an IP addresses and route for an in-band management interface using the following command format.

   portcfg mgmtif *[slot/]ge_port* [create|delete] *ipAddress netmask mtu*

2. Add the VLAN tag entry for the management interface using the following command format.

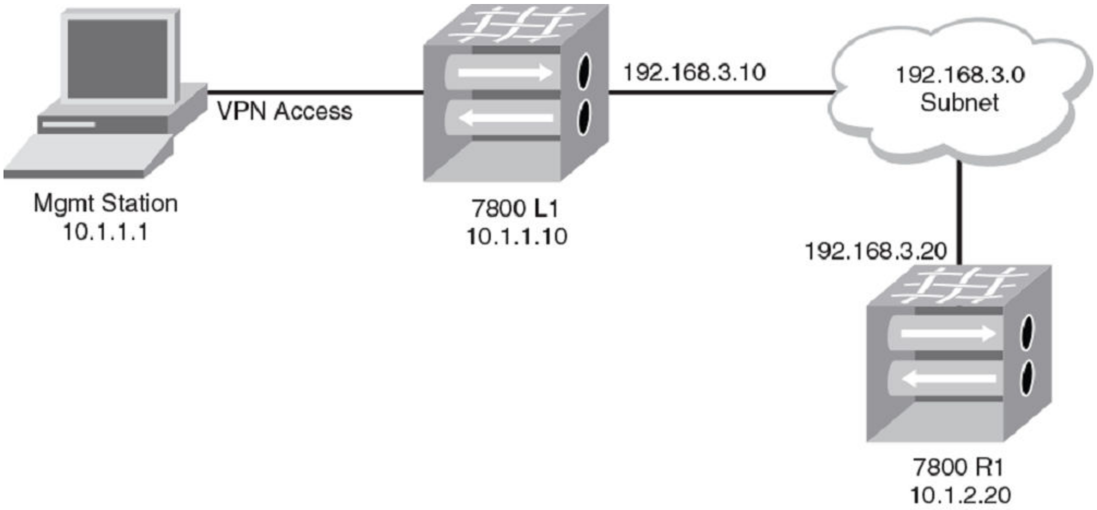
   portcfg vlantag *[slot/]ge_port* [add|delete] *ipAddress* L2COS--mgmt

IP forwarding support

IP forwarding is supported over in-band management to allow communication to the remote switch through the WAN connection. This is done by enabling IP forwarding to allow IP packets arriving at the CP interface to be forwarded through the in-band management interface to the remote side. To prevent network routing and actual bridging of the LAN side of the network to the WAN side of the network, the forwarding rules of the **ipfilter** command will default to deny any forwarding traffic. To allow forwarding, new **ipfilter** command rules must be added to specific destinations. This will prevent any unintended network traffic from being forwarded from the LAN side to the WAN side of the network.

The following figure shows an example network where the management station is located on the LAN side of 7800 L1. Using in-band management, the station can also communicate with 7800 R1.

FIGURE 22 In-band management with IPv4 forwarding



For this example, you must configure the following:

- On the management station:
  - IP address 10.1.1.1/24 (defined)
  - IP route to 192.168.3.20/32 via 10.1.1.10
- On the 7800 L1:
  - CP management address 10.1.1.10/24
  - In-band management address 192.168.3.10/24
  - IP filter forward rule with destination IP address 192.168.3.20
- On the 7800 R1:
  - CP management address 10.1.2.20/24
  - In-band management address 192.168.3.20/24
  - In-band management route to 10.1.1.1/32 via 192.168.3.10

Once all of these configurations are complete, proper IP connectivity should occur through the network. In the case where there are routed networks between the Brocade 7800 switches, you will need to add in-band management routes to each Brocade 7800 switch. Using host-specific routes will help eliminate undesired traffic. If network routes are needed, they can be substituted, but you should note that this will allow anything on that network to be forwarded, which could result in undesired disruption of FCIP traffic.

NOTE
In all routed network cases, all intermediate hops must have route entries to get to the endpoints.

Using the ipfilter command

Use the **ipfilter** command to create and manage forwarding rules for use with in-band management. For full details on this command, options, and arguments, refer to the "ipfilter" section of the *Fabric OS Command Reference*.

To create an IP forwarding rule, you must first create a new policy if one has not yet been created. The easiest way to do this is with the --clone option to create a copy of the default policy.

```
ipfilter --clone inband_ipv4 -from default_ipv4
```

A new rule can be added to allow forwarding traffic.

```
ipfilter --addrule inband_ipv4 -rule rule_number
 -dp dest_port
 -proto protocol
 -act [permit|deny] -type FWD -dip destination_IP
```

Valid *dest_port* values are any TCP or UDP port numbers or a range of port numbers that you want forwarded. Valid *protocol* values are tcp or udp . The *destination_IP* is the IP address of the in-band management interface on the remote side. After a rule is added, save the policy and activate it using the --save and --activate options of the **ipfilter** command. There can only be a single IPv4 policy active at any time. Each policy can consist of multiple rules.

WAN performance analysis tools

WAN analysis tools are designed to test connections, trace routes, and estimate the end-to-end IP path performance characteristics between a pair of Brocade FCIP port endpoints. These tools are available as options on the **portCmd** command. The following options are available:

- **portCmd** --Tperf. A tunnel test tool that generates and sends test data over an FCIP tunnel to determine the characteristics and reliability of the IP network used by the tunnel at the FCIP circuit level.
- **portCmd** --ping. Tests connections between a local Ethernet port and a destination IP address.
- **portCmd** --traceroute. Traces routes from a local Ethernet port to a destination IP address.
- **prtCmd** --wtool. Generates traffic over a pair of IP addresses to test the link for issues such as maximum throughput, congestion, loss percentage, out of order deliver, and other network conditions.
- **portShow** fcipTunnel - -perf. Displays performance statistics generated from the WAN analysis.

When using VLANs, VLAN tagging ensures that test traffic traverses the same path as real FCIP traffic. A VLAN tag entry for both the local and remote sides of the route must exist prior to issuing the **portCmd --ping** or **portCmd --traceroute** commands. Refer to Managing the VLAN tag table on page 105 for details.

The tperf option

Tperf (**portCmd --tperf**) is a utility that generates data between a local and remote switch over an FCIP tunnel. It reports the data generated and response from the remote switch to determine characteristics and reliability of the IP network used by the tunnel.

Tperf operates with a pair of Brocade 7800 switches or FX8-24 blades. One switch or blade plays the role of a data sink and the other switch or blade plays the role of the data source. During the data generation process, traffic flows from the source to the sink, then the sink responds to this traffic. The process continues for a duration that you specify with command options or until you terminate (**Ctrl** + **C**).

Normally, you should establish one Telnet or SSH session for the tperf source and one for the tperf sink. Also, open additional Telnet or SSH sessions so that you can periodically display TCP connection statistics using the -tcp or -p options of the **portshow fciptunnel** *slot/veport* command. These statistics can sometimes help you understand the tunnel bandwidth and IP network infrastructure capability.

To use Tperf, you must first create an FCIP tunnel with at least one circuit or modify an existing tunnel using the Tperf flag **-T**. As with any FCIP tunnel, this must be done on both switches. The following commands create a Tperf-enabled tunnel with a committed rate of 10000.

```
portcfg fciptunnel 16 create --remote-ip 192.168.10.1 --local-ip 192.168.10.2 10000 -
T
portcfg fciptunnel 16 create --remote-ip 192.168.10.2 --local-ip 192.168.10.1 10000 -
T
```

Tperf will test single and multiple circuit tunnels. Tperf also tests the different priority connections that are provided by an FCIP tunnel. When a Tperf-enabled tunnel is operative, it is not an active VE_Port. Fabrics will not merge over an operative FCIP Tperf tunnel. To determine if the Tperf tunnel is up, issue the following command:

```
switch:admin> portshow fciptunnel -c
-------------------------------------------------------------------------------
 Tunnel Circuit  OpStatus  Flags    Uptime  TxMBps  RxMBps ConnCnt CommRt  Met
-------------------------------------------------------------------------------
 16     -         Up      ----T-- 1h21m43s    0.00    0.00    2      -      -
 16     0 ge0     Up      ---4--s 1h21m34s    0.00    0.00    2   500/500   0
 16     1 ge1     Up      ---4--s 1h21m43s    0.00    0.00    2   500/500   0
-------------------------------------------------------------------------------
 Flags:  tunnel: c=compression m=moderate compression a=aggressive compression
                 A=Auto compression f=fastwrite t=Tapepipelining F=FICON
                 T=TPerf i=IPSec l=IPSec Legacy
 Flags: circuit: s=sack v=VLAN Tagged x=crossport 4=IPv4 6=IPv6 T=Test(CPerf)
                 L=Listener I=Initiator
```

The previous display shows VE_Port 16 as up, but a **switchshow** command for that same VE _Port will show the following:

```
switch:admin> switchshow | grep 16
 16  16   631000   --    --   Offline     VE
The Tperf command determines the path characteristics to a remote host or tunnel
destination. The syntax is as follows:
portcmd - -tperf slot/VE_port number
-sink -source
-high
|
-medium
 |
-low
-time duration
]
-unidirectional
-random
-pattern pattern
-size pdu_size
 -
interval interval
```

For full details on syntax and using this command, refer to the *Fabric OS Command Reference*.

The following examples create a Tperf data sink and a Tperf data source on VE_Port 16.

```
switch:admin> portcmd --tperf 16 -sink -interval 15
switch:admin> portcmd --tperf 16 -source -interval 15 -high -medium -low
```

Tperf generates statistics every 30 seconds by default unless you specify a different value for -interval.

TABLE 18 Tperf output

Item	Description
Tunnel ID	Numeric identifier for the Tperf tunnel.
Traffic	Priority High, Medium, or Low.
bytes tx	Number of bytes transmitted.
bytes rx	Number of bytes received.
PDUs tx	Number of protocol data units transmitted.
PDUs rx	Number of protocol data units received.
bad CRC headers rx	Number of bad CRC headers received.
bad CRC payloads rx	Number of bad CRC payloads received.
out of seq PDUs rx	Number of out-of-sequence PDUs received.
flow control count	Flow control count.
packet loss (%)	The percentage of packet loss.
bytes/second	The number of bytes transmitted per second.

TABLE 18 Tperf output (Continued)

Item	Description
last rtt	The time it took for the last round-trip between the Tperf source and the Tperf sink in milliseconds. This is calculated only on the source-side report. It is reported as N/A on the sink-side report.

Using ping to test a connection

The **portCmd** --ping command tests the connection between the IP address of a local Ethernet port and a destination IP address. If you want to use this command to test a VLAN connection when you do not have an active FCIP tunnel, you must manually add entries to the VLAN tag table on both the local and remote sides of the route, using the **portCfg vlantag** command.

The general syntax of the **portCmd** --ping command is follows:

portCmd --ping *slot/ge-port* -s *source_ip* -d *destination_ip* -n *num_request* -q *diffserv* -t *-ttl* -w *wait_time* -z *size* -v *vlan_id* -c *L2_Cos*

On the Brocade 7840 switch, because DP complexes share Ethernet ports, identification for the port is *ge n.DP n*, for example **ge0.DP0**. This directs the command to a specifc DP complex.

When using VLANs, VLAN tagging ensures that test traffic traverses the same path as real FCIP traffic. A VLAN tag entry for both the local and remote sides of the route must exist prior to issuing the **portCmd** --ping or **portCmd** --traceroute commands. Refer to Managing the VLAN tag table on page 105 for details. For details of command syntax and output examples, refer to the *Fabric OS Command Reference*.

Using traceroute

The **portCmd traceroute** command traces routes from a local Ethernet port to a destination IP address. If you want to use this command to trace a route across a VLAN when you do not have an active FCIP tunnel, you must manually add entries to the VLAN tag table on both the local and remote sides of the route using the **portCfg vlantag** command.

The general syntax of the **portCmd** --traceroute command is as follows:

portCmd --traceroute *slot/ge-port* -s *source_ip* -d *destination_ip* -h *max_hops* -f *first_ttl* -q *diffserv* -w *wait -time* -z *size* -v *vlan_id* -c *L2_Cos*

On the Brocade 7840 switch, since DP complexes share Ethernet ports, identification for the port is *ge n.DP n*, for example **ge0.DP0**. This directs the command to a specifc DP complex.

The following example traces the route between IP addresses 192.168.2.22 and 192.168.2.30 over VLAN 12 from a 7840 switch.

```
portcmd --traceroute ge2.dp1 -s 192.168.10.1 -d 192.168.20.1 -v 12
```

The following example traces the route between IP addresses 192.168.10.1 and 192.168.20.1 over VLAN 10 from an FX8-24 blade.

```
portcmd --traceroute 8/ge0 -s 192.168.10.1 -d 192.168.20.1 -v 10
```

NOTE
To trace a route with crossport addresses, refer to Using traceroute with crossports on page 36.

When using VLANS, VLAN tagging ensures that test traffic traverses the same path as real FCIP traffic. A VLAN tag entry for both the local and remote sides of the route must exist prior to issuing the **portCmd** --traceroute or **portCmd** --ping commands. Refer to Managing the VLAN tag table on page 105 for details.

For details of command syntax and output examples, refer to the *Fabric OS Command Reference*.

Using WAN Tool

WAN Tool allows you to generate traffic at a specified rate in Kbps over a pair of IP addresses to test the network link for issues such as maximum throughput, congestion, loss percentage, out of order delivery, and other network conditions. The main purpose of this tool is to determine the health of a link before deploying it for use as a circuit in an FCIP tunnel.

Following are requirements and considerations for using WAN Tool:

- WAN Tool is supported by the Brocade 7840 switch only.
- A maximum of four WAN Tool sessions are supported per DP complex.
- Each session can support a 10 Gbps connection (maximum).
- A test session can run over an IP path being used by an existing FCIP circuit between two switches; however, you must disable the circuit at each end before configuring the session.
- You must configure the WAN Tool session on the switch at each end of the circuit.
- After configuration, you can start a test from one switch only to test unidirectional traffic to the opposite switch or you can test bidirectional traffic between both switches using the **bidirectional** option. If bidirectional is specified for the test session, you can start the session at either switch.
- You can configure multiple test sessions (one per circuit) for a single port, but the total rate configured for all sessions must be equal to or less then the physical speed of the port (40 Gbps, 10 Gbps, or 1 Gbps). For example, for a 10 Gbps port, you could configure four 2.5 Gbps sessions. As another example, for a 40 Gbps interface, you could configure four 10 Gbps sessions.
- The default MTU size used in the test session is 1500, however jumbo frames are supported. Increase the MTU size for the IP address pair being tested using the **portcfg ipif** *ge_port* **create**command. For details on this command, refer to the *Fabric OS Command Reference* or Configuring an IPIF on page 65.

An FCIP tunnel and WAN Tool cannot operate at the same time since they both utilize the TCP ports 3225 and 3226. Therefore, you must disable the FCIP circuit that you are testing at the local and remote switch before you can configure a WAN Tool connection. When you configure WAN Tool on both switches with a source IP address, destination IP address, test identification, and a link rate, non-guaranteed TCP connections are established between the switches. Issuing the WAN Tool start command starts traffic flow on these connections.

Multiple non-guaranteed TCP connections are established for the WAN Tool session to insure that the traffic being generated on the circuit is as balanced as possible for the configured link rate between the TCP ports 3225 and 3226. The configured rate is split equally among 500 Mbps connections. For example, if you configure a 10 Gbps rate for the test session, twenty 500 Mbps connections are created. As another example, if you configure a 1 Gbps rate, two 500 Mbps connections are created. If the rate cannot be split equally into 500 Mbps connections, connections with different rates are created. For example, if you configure a 1.5 Gbps rate, four 375 Mbps connections are created. You can verify these connections are created after configuring WAN Tool on both switches using the **portcmd--wtool** *wt-id* **show -c** command. Refer to the example output of this command in Configuring WAN tool and displaying results on page 120.

WAN Tool commands

Configure a WAN Tool session using the **portcmd --wtool** command. The general syntax for creating a test session including all command options is as follows:

portcmd --wtool *wt-id* **create --src** *src_ip* **--dst** *dst_ip* **--rate** *link_rate* **--time** *test_time* **--bi-directional --ipsec** *policy name*.

You must configure the following parameters on each switch:

- WAN Tool session test ID (*wt-id*) - The ID doesn't have to match on each switch, but this is recommended for easier comparison of test results on both ends of the circuit when multiple test sessions are created. Valid IDs are 0 through 7.
- Link rate (*link_rate*) in Kbps - Configure the same link rate on the switch at each end of the circuit. The WAN Tool connections will not fully establish until the same rate is specified for each switch.
- IPsec policy name (*policy name*) - The policy name can be different on each switch, but must refer to the same policy configured using the **portCfg ipsec-policy** command.
- Source IP (*src_ip*) and destination IP (*dst_ip*) address - The source address will be the destination address and the destination address will be the source address on the opposite switch.
- Bi-directional (**--bi-directional**) - This is an optional parameter, but if used, configure on both switches.
- Test session time (*test_time*) - The test duration time in minutes must be configured on at least one switch, but you do not need to configure the time on both switches nor does it need to match on both switches. The test session uses the time configured on the switch where the test started. If bi-directional is specified, the session runs for the time configured on the switch where the test started, then runs for the time (if configured) configured on the opposite switch.

Modify the link rate, test time, test direction (--bi-directional) parameters, and clear statistics for a WAN Tool test session after creating a test session, using the **portcmd --wtool** *wt-id* **modify** command.

NOTE
You must stop the WAN Tool session before modifying parameters using **portcmd --wtool** *wt-id* **stop**.

Following are examples of using the **modify** parameter:

- To modify the rate, use **portcmd --wtool** *wt-id* **modify --rate** *link_rate*.
- To clear test results, use **portcmd --wtool** *wt-id* **modify --clear**.

Start and stop a configured test session on a specific switch using the following commands:

- **portcmd --wtool** *wt-id* **start**. You can specify the test duration using **portcmd --wtool** *wt-id* **start --time** *min* if the test duration has not been specified with the **create** or **modify** parameters.
- **portcmd --wtool** *wt-id* **stop**

Clear test statistics using the **portcmd --wtool** *wt-id* **modify --clear** command.

Delete test sessions using the **portcmd --wtool** *wt-id* **delete** command. Delete all configured test sessions using **portcmd --wtool all delete**. At this point, you can re-enable the circuit for operation in an FCIP tunnel using the **portCfg fcipcircuit create** command.

Display statistics from a WAN Tool session using the **portcmd --wtool** *wt-id* **show**, where *wt-id* is the ID (0-7) you used to create the test session. Display all test sessions (if multiple test sessions are configured) using the **portcmd --wtool all show**.

For more details on WAN Tool command and parameters, refer to the *Fabric OS Command Reference*.

Configuring a WAN Tool session and displaying results

1. Connect to the switch and log in using an account assigned to the admin role.
2. Disable the circuit for the IP pair that you wish to test at each switch using the **portCfg fcipcircuit modify --admin-status disable** command.

The following example disables circuit 1.

```
Switch1:admin>portCfg fcipcircuit modify 1 --admin-status disable
```

3. Verify that the circuit is disabled using the **portshow fciptunnel -c** command. The OpStatus for circuit 1 should be "Down."
4. Establish a test connection on the circuit by configuring a WAN Tool session on the switch at one end of the circuit.

   The following example configures a test connection (WAN Tool session 0) on circuit 1 between source IP of 10.1.1.1 and destination IP of 10.1.1.2.

```
Switch1:admin>portcmd --wtool 0 create --src 10.1.1.1 --dst 10.1.1.2 --rate
10000000
```

5. Configure the WAN Tool session on the switch at the other end of the circuit.

```
Switch2:admin>portcmd --wtool 0 create --src 10.1.1.2 --dst 10.1.1.1 --rate
10000000
```

   The wt-id (0) does not need to match configuration on Switch1, but this is recommended for easier comparison of test results on both ends of the circuit when multiple test sessions are created. The rate must be the same for both switches. Note that the source address of Switech1 becomes the destination address for Switch2 and the destination address becomes the source address. Refer to WAN Tool commands on page 119 for a list of WAN Tool command parameter values that must be identical for both switches in the circuit.

NOTE
The connection will not complete until a WAN Tool session is configured on both switches with an identical link rate.

6. Verify that the WAN Tool test connection has established using the **portcmd --wtool** *wt-id* **show** command and **portcmd --wtool** *wt-id* **show -c** command.

```
Switch1:admin>portcmd --wtool 0 show
wantool-id: (0)
=========================================
State              : Established
Up Time            : 7m37s
Run Time           : 0s
Time remaining     : 0s
IP Addr (L/R)      : 10.1.1.2 <-> 10.1.1.1
PMTUD              : Disabled
Comm Rate          : 10000000 Kbps (1220.70 MB/s)
Tx rate            : 4562.50 Kbps (0.56 MB/s)
Rx rate            : 4539.69 Kbps (0.55 MB/s)
Tx Utilization     : 0.05%
Rx Utilization     : 0.05%
RTT (Min/Max)      : 0.10ms/0.28ms
RTT VAR (Min/Max)  : 0.09ms/0.34ms
Local Session Statistics
  Tx pkts            : 0
Peer Session Statistics
  Rx pkts            : 0
  Ooo pkts           : 0
  Drop pkts          : 0 (0.00%)
Switch1:admin>portcmd --wtool 0 show -c
Id      Port(L/R)         Rate(Tx/Rx)         UpTime    RunTime
=======================================================================
6     63494 / 3225       0.03 / 0.03          8m8s         0s
17    63490 / 3225       0.03 / 0.03          8m8s         0s
14    63498 / 3225       0.03 / 0.03          8m8s         0s
3     61443 / 3226       0.03 / 0.03          8m8s         0s
11    61447 / 3226       0.03 / 0.03          8m8s         0s
9     61446 / 3226       0.03 / 0.03          8m8s         0s
1     61442 / 3226       0.03 / 0.03          8m8s         0s
20    63491 / 3225       0.03 / 0.03          8m8s         0s
8     63495 / 3225       0.03 / 0.03          8m8s         0s
12    63497 / 3225       0.03 / 0.03          8m8s         0s
4     63493 / 3225       0.03 / 0.03          8m8s         0s
16    63489 / 3225       0.03 / 0.03          8m8s         0s
13    61448 / 3226       0.03 / 0.03          8m8s         0s
19    61440 / 3226       0.03 / 0.03          8m8s         0s
5     61444 / 3226       0.03 / 0.03          8m8s         0s
```

```
15    61449 / 3226       0.03 / 0.03          8m8s          0s
7     61445 / 3226       0.03 / 0.03          8m8s          0s
18    61441 / 3226       0.03 / 0.03          8m8s          0s
10    63496 / 3225       0.03 / 0.03          8m8s          0s
2     63492 / 3225       0.03 / 0.03          8m8s          0s
======================================================================
Number of Connections:20
```

The example output from the **--wtool 0 show** indicates that the connection has an established state. The example output from the **--wtool 0 show -c** command displays the TCP Lite connections created between TCP ports 3225 and 3226 to balance the test traffic. For the 10 Gbps test connection, twenty TCP Lite connections are created.

7. If you have created multiple WAN Tool sessions, you can verify basic connection information using the **--wtool all show** command.

```
Switch1:admin>portcmd --wtool all show
Id  CommRate     Port     Local IP <-> Remote IP    TxMBps   RxMBps   Drop %
=========================================================================
0    10000000   ge9.dp0   10.1.1.2<->10.1.1.1       0.56     0.55     0.00
4    10000000   ge9.dp1   10.1.2.2<->10.1.2.1       0.56     0.56     0.00
=========================================================================
```

Output for this example shows that WAN Tool session 0 was created to test the circuit with IP address pair 10.1.1.2 and 10.1.1.1 and session 4 was created testing the circuit with IP address pair 10.1.2.2 and 10.1.2.1.

8. Start traffic on the test connection by entering the **portcmd --wtool** *wt-id* **start** command. If you did not specify a test duration with the **prtcmd --wtool** *wt-id* **create** command, you can do so with the start command using the **-time** option.

```
Switch1:admin>portcmd --wtool 0 start -time 10
```

9. Verify that the test session started by entering the **portcmd --wtool** *wt-id* **show** command.

```
Switch1:admin>portcmd --wtool 0 show
State               : Running
Up Time             : 15m39s
Run Time            : 6s
Time remaining      : 9m54s
IP Addr (L/R)       : 10.1.1.1 <-> 10.1.1.2
PMTUD               : Disabled
Comm Rate           : 10000000 Kbps (1220.70 MB/s)
Tx rate             : 9394147.00 Kbps (1146.75 MB/s)
Rx rate             : 6102.34 Kbps (0.74 MB/s)
Tx Utilization      : 93.94%
Rx Utilization      : 0.06%
RTT (Min/Max)       : 2.23ms/3.00ms
RTT VAR (Min/Max)   : 0.21ms/1.37ms
Local Session Statistics
  Tx pkts             : 4766856
Peer Session Statistics
  Rx pkts             : 0
  Ooo pkts            : 0
  Drop pkts           : 0 (0.00%)
```

Note that the "State" shows that the test is running and other statistics display as well, such as test "Run Time" and "time remaining".

10. Start the test from the other switch by entering the **portcmd --wtool** *wt-id* **start --time** *test_time*command.

NOTE
If you used the **bi-directional** option when creating the session, you can start the session on either switch.

```
Switch2:admin>portcmd --wtool 0 start -time 10
```

11. Verify that the test session started on the other switch by entering the **portcmd --wtool** *wt-id* **show** command.

```
Switch2:admin>portcmd --wtool 0 show
```

12. Delete the WAN Tool session on both switches using the **portcmd --wtool** *wt-id* **delete** command.

13. To verify that the WAN Tool session is disabled, enter the **portcmd --wtool** *wt-id* **show** command.
14. Enable the circuit from each switch using the **portcfg fcipcircuit** *port* **create** command. The following example enables the circuit from Switch1.

```
Switch1:admin>portcfg fcipcircuit 12 create 1 --remote-ip 10.1.1.2 --local-ip
10.1.1.1 -b 15500 -B 62000
```

Resolving test session problems

If output from the **portcmd --wtool** *wt-id* **show** command shows that the "State" is down, constantly in progress, or the connection times out (changes from an up to down state) the WAN Tool test connection is not being established. Verify that you have configured the session on both switches in the circuit with appropriate parameters and values. Refer to the list of parameters required for each switch in WAN Tool commands on page 119.

Common problems in establishing a connection can result from the following WAN Tool configuration problems:

- The test rate doesn't match on each switch.
- The test rate on a single circuit or multiple circuits on a port is greater than the rate allowed for the port. Note that this will generate a warning that the bandwidth has been exceeded and blocks you from creating a session.
- The IPsec policy doesn't match on each switch. Note that the IPsec names do need to match, but must the names must refer to the same policy.
- Configured source and destination IP addresses are not correct on one or both switches.
- The wt-id is not 0-7.

Using the portshow command

Use the **portshow** command to display port operational information on Brocade extension switches and blades. The *Fabric OS Command Reference* provides complete descriptions of the **portshow** command syntax and options. The following sections identify a few specific outputs that may be useful for maintenance and troubleshooting.

Displaying IP interfaces

The following example displays IP interface information for a Brocade 7800 or a Brocade 7840 switch.

```
switch:admin> portshow ipif ge0
```

The following example displays IP interface information for an FX8-24 blade.

```
switch:admin> portshow ipif 1/xge0
```

Displaying IP routes

The following example displays IP route information for a Brocade 7800 or a Brocade 7840 switch.

```
switch:admin> portshow iproute ge5
```

The following example displays IP route information for an FX8-24 blade.

```
switch:admin> portshow iproute 1/xge0
```

Displaying FCIP tunnel information

The following example of the **portshow fciptunnel** command is used most often to determine FCIP tunnel status.

```
switch:admin> portshow fciptunnel all -c
```

Displaying IP addresses

You can display IP addresses configured for specific circuits using the ip-address option with the *circuit* option as in the following example.

```
switch:admin> portshow fciptunnel all --circuit --ip-address
```

Displaying performance statistics

Display a summary of performance statistics for tunnels and circuits using the circuit, perf, and summary options as in the following example.

```
switch:admin> portshow fciptunnel all --circuit --perf --summary
```

Displaying QoS statistics

Display QoS statistics for tunnels using the **--qos** and **--summary** and options as in the following example.

```
switch:admin> portshow fciptunnel all --qos --summary
```

Displaying details

You can display configuration details using the detail option with the all option as in the following example.

```
switch:admin> portshow fciptunnel all --detail
```

Displaying FCIP tunnel information

The following example will display general tunnel information related to port 16 on a Brocade 7800 switch.

```
switch:admin> portshow fciptunnel 16
```

Displaying an FCIP tunnel with FCIP circuit information

The following example adds circuit information to the **portshow fciptunnel** command output using the -c option.

```
switch:admin> portshow fciptunnel 17 -c
```

Displaying FCIP tunnel performance

The following example will display performance statistics for a tunnel associated with port 17 on a Brocade 7800 switch.

```
switch:admin> portshow fciptunnel 17 --perf
```

Displaying FCIP tunnel TCP statistics

The following example will display TCP connections for a tunnel associated with port 17 on a Brocade 7800 switch.

```
switch:admin>portshow fciptunnel 17 -c --tcp
```

You can reset statistics counters to zero to display only new statistics with the --tcp option from the time you issue the reset using the following command.

```
switch:admin> portshow fciptunnel 17 -c --tcp --reset
```

You can display the entire lifetime of statistics for the tunnel using the following command. The time basis for the statistics will display in the output.

```
switch:admin> portshow fciptunnel 17 -c --tcp --lifetime
```

Displaying FCIP circuits

The following example will display all FCIP circuit information.

```
switch:admin> portshow fcipcircuit all
```

Displaying a single circuit

The following example will display information for circuit 1 on tunnel 16 of a Brocade 7800 switch.

```
switch:admin> portshow fcipcircuit 16 1
```

Displaying TCP statistics for circuits

The following example displays TCP statistics for circuits associated with VE_Port 12 of an FX8-24 blade.

```
switch:admin> portshow fcipcircuit 3/12 --tcp
```

You can reset statistics counters to zero to display only new statistics with the --tcp option from the time you issue the reset using the following command.

```
switch:admin> portshow fcipcircuit 3/12 -tcp --reset
```

You can display the entire lifetime of statistics for the circuit using the following command. The time basis for the statistics will display in the output.

```
switch:admin> portshow fcipcircuit 3/12 -tcp --lifetime
```

Displaying FCIP circuit performance

The following example will display FCIP circuit performance information for circuit 1 on tunnel 20.

```
switch:admin> portshow fcipcircuit 20 1 --perf
```

Displaying QoS prioritization for a circuit

The following example will display QoS prioritization for FCIP circuit 1 on tunnel 20 of a Brocade 7800 switch.

```
switch:admin> portshow fcipcircuit 20 1 --perf --qos
```

Displaying FCIP tunnel information (FX8-24 blade)

You can use the **portShow fcipTunnel** command to view the performance statistics and monitor the behavior of an online FCIP tunnel. The command syntax is as follows.

The following example shows **portShow fcipTunnel** with the --c option to display FCIP circuits of tunnel 0.

```
switch:admin06> portshow fciptunnel 8/12 0 —c
```

FCIP tunnel issues

The following are common FCIP tunnel issues and recommended actions for you to follow to fix the issue.

FCIP tunnel does not come online

Troubleshoot this issue using the following steps.

1. Confirm Ethernet port is online.

   ```
   portshow ge1
   Eth Mac Address: 00.05.1e.37.93.06
   Port State: 1   Online
   Port Phys:  6   In_Sync
   Port Flags: 0x3  PRESENT ACTIVE
   Port Speed: 1G
   ```
2. Confirm the IP configuration is correct on both tunnel endpoints using the following command.

   ```
   portshow ipif ge1
   ```
3. Enter the **portCmd** --ping command to the remote tunnel endpoint from both endpoints.

   The -s value is the source IP address; the -d value is the destination IP address.

   ```
   portcmd --ping ge1 -s 11.1.1.1 -d 11.1.1.2
   ```

   If the command is successful, then you have IP connectivity and your tunnel should come up. If not, continue to the next step.

   When using VLANS, VLAN tagging ensures that test traffic traverses the same path as real FCIP traffic. A VLAN tag entry for both the local and remote sides of the route must exist prior to issuing the **portCmd** --ping or **portCmd** --traceroute commands. Refer to Managing the VLAN tag table on page 105 for details.
4. Enter the **portCmdp** --traceroute command to the remote tunnel endpoint from both endpoints.

   ```
   portcmd --traceroute ge1 -s 11.1.1.1 -d 11.1.1.2
   ```

5. If there are routed IP connections that provide for the FCIP tunnel, confirm that both ends of the tunnel have defined IP routes. The tunnel or route lookup may fail to come online because of a missing but required IP route.

   Refer to the Configuring an IP route on page 66 to review the setup of the IP route.
6. Confirm the FCIP tunnel is configured correctly using the following command.

   ```
   portshow fciptunnel all
   ```

   Confirm that the compression, FastWrite, and OSTP settings match at each endpoint or the tunnel may not come up. Confirm that the local and destination IP address and WWN are accurate.
7. Generate an Ethernet sniffer trace.

   Rule out all possible blocking factors. Routers and firewalls that are in the data path must be configured to pass FCIP traffic (TCP port 3225) and IPsec traffic, if IPsec is used (UDP port 500). If possible blocking factors have been ruled out, simulate a connection attempt using the **portCmd** --ping command, from source to destination, and then generate an Ethernet trace between the two endpoints. The Ethernet trace can be examined to further troubleshoot the FCIP connectivity.

FCIP tunnel goes online and offline

An FCIP tunnel that goes online and then online (bouncing tunnel) is a common problem. This usually occurs because of an overcommitment of available bandwidth resulting in the following behaviors:

- Too much data tries to go over the link.
- Management data gets lost, queued too long, and timeouts expire.
- Data times out multiple times.

Take the following steps to gather information.

1. Verify what link bandwidth is available.
2. Confirm the IP path is being used exclusively for FCIP traffic.
3. Confirm that traffic shaping is configured to limit the available bandwidth using the following command.

   ```
   portShow fciptunnel all -tcp
   ```

   Examine data from both routers. This data shows retransmissions indicating input and output rates on the tunnels.
4. For the 7800 switch and FX8-24 blade, run **Tperf** command to gather WAN performance data. For the 7840 switch, use the WAN tool.

FCIP links

The following list contains information for troubleshooting FCIP links:

- When deleting FCIP links, you must delete them in the exact reverse order in which they were created. That is, first delete the tunnels, then the IP interfaces, and finally the port configuration. Statically defined IP routes are not removed automatically and must be removed manually before deleting IP addresses.
- IP addresses and FCIP configurations are retained by slot in the system.
- The **portCmd** --ping command only verifies physical connectivity. This command does not verify that you have configured the ports correctly for FCIP tunnels.
- Ports at both ends of the tunnel must be configured correctly for an FCIP tunnel to work correctly. These ports can be either VE_Ports or VEX_Ports. A VEX_Port must be connected to a VE_Port.

- When configuring routing over an FCIP link for a fabric, the edge fabric will use VE_Ports and the backbone fabric will use VEX_Ports for a single tunnel.
- If an FCIP tunnel fails with the "Disabled (Fabric ID Oversubscribed)" message, the solution is to reconfigure the VEX_Port to the same fabric ID as all of the other ports connecting to the edge fabric.
- Because of an IPsec RASLog limitation, you may not be able to determine an incorrect configuration that causes an IPsec tunnel to not become active. This misconfiguration can occur on either end of the tunnel. As a result, you must correctly match the encryption method, authentication algorithm, and other configurations on each end of the tunnel.

Gathering additional information

The following commands should be executed and their data collected before the **supportSave** command is run. Using the **supportSave** command can take ten minutes or more to run, and some of the information is time critical.

- **traceDump** -n
- **portTrace** --show all
- **portTrace** --status

For issues specific to tunnel ports, run and collect the data from the following commands:

- **slotShow**
- **portShow** *slot/ge_port*

If possible, run and collect the data from the following commands:

- **portShow ipif** all *slot/ge_port*

- **portShow arp** all *slot/ge_port*
- **portShow iproute** all *slot/ge_port*
- **portShow fciptunnel** *slot/ge_port* all|tunnel ID

- **portShow fciptunnel** all --perf
- **portShow fciptunnel** all -c
- **portShow fciptunnel** all --circuit --perf --summary
- **portShow fciptunnel** all --circuit --perf --tcp --qos
- **portCmd** --ping --traceroute --perf
- **portCmd** --ping
- **portCmd traceroute**

Finally, gather the data from the **supportsave** command.

Refer to the *Fabric OS Administrator's Guide* or *Fabric OS Command Reference* for complete details on these commands.

Using FTRACE

FTRACE is a support tool used primarily by your switch support provider. FTRACE can be used in a manner similar to that of a channel protocol analyzer. You can use FTRACE to troubleshoot problems through a Telnet session rather than using an analyzer or sending technical support personnel to the installation site.

CAUTION

FTRACE is meant to be used solely as a support tool and should be used only by Brocade support personnel, or at the request of Brocade support personnel. The FTRACE command is restricted to the root switch user.

FTRACE is always enabled on extension switches and blades, and the trace data is automatically captured.

FTRACE configuration

A default configuration for FTRACE is provided for each of the two FCIP DP complexes on the Brocade the FX8-24 blade and 7840 switch and for the single Brocade 7800 FCIP DP complex. This allows tracing of events related to the FCIP complexes.

You can use the root **portcfg ftrace** *slot/ge_port* cfg command to change FTRACE configuration settings as described in Configuring IP addresses and routes on page 110.

Brocade 7800 switch and FX8-24 blade

The default configuration creates four FTRACE buffers of 100,000 trace events that will be used until a trigger event (programmed trigger point in the FCIP logic) occurs. Trigger events include unexpected events or events that include FC abort sequences or other errors when FCIP emulation features are enabled on the tunnel.

The default configuration does not allow re-use of a trace buffer that includes one or more trigger events. The FTRACE configuration item that controls this function is called Auto Checkout (ACO). The default configuration of FTRACE provides for capturing, at a minimum, the first four error time periods in the four FTRACE buffers. That is because the default setting has enabled FTRACE ACO processing. When a buffer is checked out, it will not be reused until it is manually checked in or cleared through the supportsave process.

If the FTRACE configuration is changed so that ACO is disabled, then instead of post-filling and then checking out, the buffer is marked as triggered. If multiple trigger events subsequently occur so that all buffers are marked triggered, FTRACE will find the oldest triggered buffer and make it the current buffer. In this configuration, FTRACE will be set up to capture the last three error time periods.

FTRACE data contents are included in a switch supportsave capture. After the supportsave has been captured, the FTRACE buffers will be reset and all buffers that were previously either "checked out" or "triggered" return to an "unused" state.

Change the FTRACE ACO configuration using the following root command:

```
portcfg ftrace [slot/]vePort cfg
```

Refer to Changing configuration settings on page 130 for more information.

Brocade 7840 switch

FTRACE has been enhanced on the Brocade 7840 to allow more trace saving options than for the Brocade 7800 switch or FX8-24 blade. The default FTRACE configuration has been changed on this platform as a result of those enhancements. For a display of the default configuration for the Brocade 7840 using the **portshow ftrace** *ve_port* stats command, refer to Displaying FTRACE status on 7840 switch on page 133.

The Brocade 7840 includes two FCIP Data Processing (DP) complexes. Each DP complex has an FTRACE instance. The default configuration for FTRACE on the Brocade 7840 defines eight FTRACE buffers for FCIP trace events on each DP complex. The default configuration defines 300,000 trace entries (trace records) per trace buffer. The default FTRACE configuration enables auto checkout (ACO) for the first four buffers and disables ACO for the last four. The Brocade 7840 switch has a solid state disk (SSD) file system in each DP complex. This can be used to save copies of triggered FTRACE buffers. Use of the SSD to save FTRACE buffers is enabled by default and by the "Save to Flash" **portcfg ftrace** *ve_port* cfg command.

On the Brocade 7840, you can enable ACO for each defined FTRACE buffer. FTRACE processing varies when the FTRACE buffer is defined with ACO enabled or disabled.

ACO enabled - If the FTRACE buffer is defined with ACO enabled, when that buffer is the "current" FTRACE buffer and a trigger event occurs, FTRACE will post fill that buffer to the end (or add the post fill percentage of more trace entries). When the post filling process is occurring the FTRACE buffer state will be reported as "post fill". When the post filling process has completed, the buffer state will be reported as "checked out," and the next sequential available buffer number will be assigned to the current buffer (state "current"). If all FTRACE buffers are marked as "checked out, "FTRACE will no longer be recording trace entries. The default configuration therefore will capture at least the first four error traces, permanently check out those buffers, and then move them to the ACO-off buffers. FTRACE buffers that have been checked out will be saved in a supportsave capture. When the supportsave is complete, the buffers will return to an "unused" state and will be available for new traces. You can use the **portshow ftrace** *ve_port cmd* command to check in a checked out buffer.

ACO disabled - If the FTRACE buffer is defined with ACO disabled, when that buffer is the "current" FTRACE buffer and a trigger event occurs, FTRACE processing will complete the same post filling process as described above. When completed, if the "Save to Flash" configuration option was enabled, the buffer will move to a "saving" state, and the next available buffer will be made as the current trace buffer. The Brocade 7840 will save as many as eight FTRACE buffers in the DP SSD file system. If there are already eight saved FTRACE buffers in the file system, the oldest trace buffer will be replaced by the current buffer being saved. When the save-to-flash processing completes, the buffer will be marked as "triggered". If the "Save to Flash" option is not enabled, the buffer will be immediately marked as "triggered" and the next sequentially available FTRACE buffer will be marked as the "Current" buffer.

In the default configuration, FTRACE will therefore capture at least the first four error events (in buffers 1, 2, 3 and 4). It will capture the last three error events in triggered buffers (5-7) and will always have a current buffer. Buffers 5-7 will also potentially have as many as 10 saved prior trigger events reported and saved in the DP SSD file system.

FTRACE data contents are included in a switch supportsave capture. After the supportsave has been captured, the FTRACE buffers will be reset and all buffers that were previously either "Checked Out" or "Triggered" return to an "unused" state.

Change the FTRACE ACO configuration using the root **portcfg ftrace** *ve_port* cfg command. Refer to Changing configuration settings on page 130 for more information.

Changing configuration settings

Use the root **portcfg ftrace** *slot/ge_port* cfg command to change FTRACE configuration settings. The configuration for FTRACE is defined using the first VE_Port on the switch or blade DP complex as follows:

- Brocade 7800 switch - VE_Port 16
- Brocade FX8-25 blade - VE_Port 22 on DP0 and VE_Port 12 on DP1
- Brocade 7840 switch - VE_Port 21 on DP0 and VE_Port 34 on DP1

To change FTRACE configuration settings on a Brocade 7800 switch, if applicable, set the context where VE port 16 is defined, and then issue the following command as root user only:

```
portcfg ftrace 16 cfg
```

To change FTRACE configuration settings on the first FCIP DP complex (DP0) on a Brocade 7840 switch, if applicable, set the context where VE_ Port 24 is defined, and then issue the following command as the root user only:

```
portcfg ftrace 24 cfg
```

To change FTRACE configuration settings on the first FCIP DP complex (DP0) on a Brocade FX8-24 blade, if applicable, set the context where the VE_Port 22 is defined, and then issue the following command as the root user only:

```
portcfg ftrace slot_number/22 cfg
```

To change FTRACE configuration settings on the second FCIP DP complex on a Brocade FX8-24 (DP1), if applicable, set the context to where VE port 12 is defined, and then issue the following command as the root user only:

```
portcfg ftrace slot_number/12 cfg
```

To change FTRACE configuration settings on the first FCIP DP complex (DP0) on a Brocade 7840 switch, if applicable, set the context where the VE_Port 24 is defined, and then issue the following command as the root user only:

```
portcfg ftrace slot_number/24 cfg
```

To change FTRACE configuration settings on the second FCIP DP complex (DP1) on a 7840 switch, if applicable, set the context to where VE port 34 is defined, and then issue the following command as the root user only:

```
portcfg ftrace slot_number/12 cfg
```

Note that **portcfg** is an interactive command sequence and will prompt you for configuration items.

Brocade 7840 switch example

Following is an example of the interactive command sequence that illustrates where you are prompted to change FTRACE configuration settings on a Brocade 7840 switch. To change the settings, set the context where VE_Port 34 is defined, and then issue the **portcfg ftrace 34 cfg** command as root user only.

NOTE
User input lines in following example of this interactive command have been annotated to help you select configuration options. Those notes in italic font, such as **Enables FTRACE (default is y)**, indicate options that you can modify. Those in bold font, such as as ***Sets the trace mask***, indicate options that you should not modify without direction from a support representative.

```
switch_10:FID10:root> portcfg ftrace 34 cfg

  ***  FTRACE INTERACTIVE CONFIGURATION  ***

  ***  Note: A reboot is necessary to    ***
  ***  activate a change in the number   ***
  ***  of buffers or records.            ***
Enable FTRACE?                      (Y,y,N,n): [y] y          *Enables FTRACE -default y*
Buffers                             (0-16): [8]               *Sets number of trace buffers -default 8*
Records (decimal, no commas)        (0-262,144): [300,000]    *Sets  number of trace records per buffer
-default 200,000*
Auto Checkout?                      (Y,y,N,n): [y]            *Enables ACO (default y)*
Auto Checkout is on, config at least 1 buffer accordingly.
    Auto Checkout buffer 0          (Y,y,N,n): [y]            *Enables ACO for buffer 0 -default y*
    Auto Checkout buffer 1          (Y,y,N,n): [y]            *Enables ACO for buffer 1 -default y*
    Auto Checkout buffer 2          (Y,y,N,n): [y]            *Enables ACO for buffer 2 -default y*
    Auto Checkout buffer 3          (Y,y,N,n): [y]            *Enables ACO for buffer 3 -default y*
    Auto Checkout buffer 4          (Y,y,N,n): [n]            *Disables ACO for buffer 4 -default n*
    Auto Checkout buffer 5          (Y,y,N,n): [n]            *Disables ACO for buffer 5 -default n*
    Auto Checkout buffer 6          (Y,y,N,n): [n]            *Disables ACO for buffer 6 -default n*
    Auto Checkout buffer 7          (Y,y,N,n): [n]            *Disables ACO for buffer 7 -default n*
Save to Flash?                      (Y,y,N,n): [y]            *Enables saving non-ACO to flash -
default y*
Post Percentage (decimal)           (0-100): [5]              *Sets the post fill percentage -default
5*
Trace Mask (*)                      (0-fffffff): [8000dffb]   *Sets the trace mask -default 8000dffb*
Trigger Mask (T)                    (0-fffffff): [1]          *Sets the trigger mask -default 1*
Display Mask (-)                    (0-ffffffff): [ffffffff]  *Sets the trace display mask -default
ffffffff*
Enable VE Traces?                   (Y,y,N,n): [y]            *Enables VE event traces -default y*
Enable FCIP Traces?                 (Y,y,N,n): [y]            *Enables FCIP event traces -default y*
Enable TCPIP Traces?                (Y,y,N,n): [y]            *Enables TCP/IP event traces -default y*
Enable TCPIP Conn Traces?           (Y,y,N,n): [y]            *Enables TCP/IP Connection event traces -
default y*
Enable IP Traces?                   (Y,y,N,n): [y]            *Enables IP Event traces -default y*
Enable ARL Traces?                  (Y,y,N,n): [y]            *Enables ARL Event traces -default y*
Enable Ethernet Traces?             (Y,y,N,n): [n]            *Disables Ethernet traces -default n*
Enable IP API Traces?               (Y,y,N,n): [y]            *Enables IP/API even traces -default y*
Enable FCIP MSG Traces?             (Y,y,N,n): [y]            *Enables FCIP Msg traces -default y*
Enable VDM Traces?                  (Y,y,N,n): [n]            *Disables VDM traces -default n*
Configuration complete.
Operation Succeeded
switch_10:FID10:root>

To correctly and completely delete an FTRACE configuration and reset to defaults, perform the
following command sequences:
switch_10:FID10:root> portcfg ftrace 34 del

  *** Note: This command will clear out  ***
  *** the current config and FTRACE will ***
  *** be reset to default values.        ***

Do you wish to continue?           (Y,y,N,n): [n] y

Operation Successful

switch_10:FID10:root> reboot
/* After switch completes reboot sequence */
switch_10:FID10:root> portcfg ftrace 34 cfg
/* repeat the configuration or leave as default */
```

Brocade 7800 switch example

Following is an example of the interactive command sequence that illustrates where you are prompted to change FTRACE configuration settings on a Brocade 7800 switch. To change the settings, set the context where VE_Port 16 is defined, and then issue the **portcfg ftrace 16 cfg** command as root user only.

```
switch6:root> portcfg ftrace 16 cfg
  ***  FTRACE INTERACTIVE CONFIGURATION  ***
  ***  Note: A reboot is necessary to    ***
  ***  activate a change in the number   ***
  ***  of buffers or records.            ***
  Setting up ftrace configuration defaults.
Enable FTRACE?                      (Y,y,N,n): [n] y
Auto Checkout?                      (Y,y,N,n): [n] y or n
Buffers                             (0-6): [4] 6
Records (decimal, no commas)        (0-349,520): [100,000] 120000
Post Percentage (decimal)           (0-100): [5] 6
Reference the table below to set the TRACE, TRIGGER, and DISPLAY masks
 *-Bit 31 [0x80000000]: Software Structure
  -Bit 19 [0x00080000]: EtRX - Ethernet Received Frame
  -Bit 18 [0x00040000]: EtSX - Ethernet Send Frame to Peer
  -Bit 17 [0x00020000]: TnTX - Tunnel Received Peer Frame
  -Bit 16 [0x00010000]: TnSX - Tunnel Send Frame to Peer
  -Bit 15 [0x00008000]: FcT - FC FWD Frame From Peer
  -Bit 14 [0x00004000]: FcR - FC FWD Received Frame
  -Bit 13 [0x00002000]: Dsc - Discarded Frame
  -Bit 12 [0x00001000]: Data - Frame Data
 *-Bit 11 [0x00000800]: State Change
 *-Bit 10 [0x00000400]: CpRX - Frame Received From CP
  -Bit  9 [0x00000200]: CpSX - Frame Sent To CP
 *-Bit  8 [0x00000100]: ToP - Sent To Peer
  -Bit  7 [0x00000080]: Tfx - Emulation FC Frame From Peer
 *-Bit  6 [0x00000040]: Rfx - Emulation FC Received Frame
 *-Bit  5 [0x00000020]: Sfx - Send Frame
 *-Bit  4 [0x00000010]: Gfx - Generated Frame
 *-Bit  3 [0x00000008]: FC SOFi1/2/3 or Class F Frames
  -Bit  2 [0x00000004]: FC SOFn1/2/3 Frames
T*-Bit  1 [0x00000002]: Msg - Information
T*-Bit  0 [0x00000001]: Err - Error
Trace Mask (*)                      (0-fffffff): [8000fefb]
Trigger Mask (T)                    (0-fffffff): [1]
Display Mask (-)                    (0-fffffff): [8000fefb]
Enable VE Traces?                   (Y,y,N,n): [y]
Enable FCIP Tunnel Traces?          (Y,y,N,n): [y]
Enable TCPIP Traces?                (Y,y,N,n): [y]
Enable TCPIP Conn Traces?           (Y,y,N,n): [n]
Enable IP Traces?                   (Y,y,N,n): [n]
Enable ARL Traces?                  (Y,y,N,n): [n]
Enable Ethernet Traces?             (Y,y,N,n): [n]
Enable IP API Traces?               (Y,y,N,n): [n]
Enable FCIP MSG Traces?             (Y,y,N,n): [n]
Enable VDM traces?                  (Y,y,N,n): [n]
Operation Succeeded
spike64:root>
```

Displaying FTRACE status on an FCIP DP complex

To display the current FTRACE status on an FCIP DP complex, issue the following command as the root user:

```
portshow ftrace [slot/]vePort stats
```

The *vePort* is in the current logical switch context.

Brocade 7840 switch example

Following is an example of displaying FTRACE status using the **portshow ftrace** *slot/ve_port* stats command. Note that this is the default configuration for the Brocade 7840 switch.

```
skybolt63:FID128:root> portshow ftrace 43 stats

VE traces:          On-all        Trace Mask:        0x8000dffb (*)
FCIP Tunnel traces: On-all        Trigger Mask:      0x00000001 (T)
TCPIP traces:       On-all        Display Mask:      0xffffffff (-)
TCPIP Conn. traces: On-all        Tunnel Mask:       Inactive
IP traces:          On-all        Post trigger:      5% - 10000 events
ARL traces:         On-all        Record Size:       128
ETHERNET traces:    Off           Save to Flash:     Enabled
IP API traces:      On-all        FTRACE is:         Enabled
FCIP MSG traces:    On-all        Debug level:       4-Normal (low)
VDM traces:         Off           CLIB / HAL:        Off / Off

*-Bit 31 [0x80000000]: Software Structure
  -Bit 19 [0x00080000]: EtRX - Ethernet Received Frame
  -Bit 18 [0x00040000]: EtSX - Ethernet Send Frame to Peer
  -Bit 17 [0x00020000]: TnTX - Tunnel Received Peer Frame
  -Bit 16 [0x00010000]: TnSX - Tunnel Send Frame to Peer
*-Bit 15 [0x00008000]: FcT - FC FWD Frame From Peer
*-Bit 14 [0x00004000]: FcR - FC FWD Received Frame
  -Bit 13 [0x00002000]: Dsc - Discarded Frame
*-Bit 12 [0x00001000]: Data - Frame Data
*-Bit 11 [0x00000800]: State Change
```

```
*-Bit 10 [0x00000400]: CpRX - Frame Received From CP
*-Bit  9 [0x00000200]: CpSX - Frame Sent To CP
*-Bit  8 [0x00000100]: ToP - Sent To Peer
*-Bit  7 [0x00000080]: Tfx - Emulation FC Frame From Peer
*-Bit  6 [0x00000040]: Rfx - Emulation FC Received Frame
*-Bit  5 [0x00000020]: Sfx - Send Frame
*-Bit  4 [0x00000010]: Gfx - Generated Frame
*-Bit  3 [0x00000008]: FC SOFi1/2/3 or Class F Frames
  -Bit  2 [0x00000004]: FC SOFn1/2/3 Frames
*-Bit  1 [0x00000002]: Msg - Information
T*-Bit  0 [0x00000001]: Err - Error

+-----+----------+-----+--------+------------+-----+------+------+-------+--------+
|     |          |     |        |Trace Header|Wrap | In   | Out  |Switch | Switch |
|  Id | State    | ACO |  Size  |  Address   |Count| OXID | OXID |  Date |  Time  |
+-----+----------+-----+--------+------------+-----+------+------+-------+--------+
|   0 |  Current |  on | 200000 | 0x0b0f7480 |   0 | FFFF | FFFF |       |        |
|   1 |   unused |  on | 200000 | 0x0b0f7780 |   0 | FFFF | FFFF |       |        |
|   2 |   unused |  on | 200000 | 0x0b0f7a80 |   0 | FFFF | FFFF |       |        |
|   3 |   unused |  on | 200000 | 0x0b0f7d80 |   0 | FFFF | FFFF |       |        |
|   4 |   unused | off | 200000 | 0x0b0f8080 |   0 | FFFF | FFFF |       |        |
|   5 |   unused | off | 200000 | 0x0b0f8380 |   0 | FFFF | FFFF |       |        |
|   6 |   unused | off | 200000 | 0x0b0f8680 |   0 | FFFF | FFFF |       |        |
|   7 |   unused | off | 200000 | 0x0b0f8980 |   0 | FFFF | FFFF |       |        |
+-----+----------+-----+--------+------------+-----+------+------+-------+--------+
```

The table at the bottom of the output example has the following columns:

- Id - The FTRACE trace buffer identifier or buffer number.
- State - The FTRACE buffer state for that buffer number. The state can be one of the following:
  - Current - The buffer is the current active buffer in use for FCIP events.
  - Triggered - The buffer has been used to record an error event from the FCIP complex. This state is used only when the Auto Checkout option was disabled.
  - Checked Out - The buffer has been used to record an error event from the FCIP complex, and the buffer will not be overwritten.
  - Post Fill - A trigger event has been encountered, and the FTRACE buffer is currently being post-filled with a number of post-error events. Once the post-filling has been completed, the buffer will transition to either a "Checked Out" or "Triggered" state.
  - Unused -The buffer has not been used to capture any FCIP events. The buffer will be used when the prior buffer in the list transitions to either a "Checked Out" or "Triggered" state.
- ACO - Auto Checkout enabled (on) or disabled (off) status.
- Size - The number of trace records that are in the buffer.
- Trace Header Address - A memory address used internally for controlling access to the trace buffer.
- Wrap Count - The number of times that a trace buffer has been wrapped. The trace is a circular buffer that wraps after the size number of trace events has been exceeded.
- In OXID and Out OXID - Not used until the buffer is being analyzed.
- Switch Date - Indicates the system date when the buffer transitioned to either a "Checked Out" or "Triggered" state.

Brocade 7800 switch or FX8-24 blade example

Following is an example of displaying FTRACE status using the **portshowftrace** *slot*/*ve_port* **stats** command.

```
Slot 0:
VE traces (0-31): (0xffffffff)  On      Trace Mask:    0x8000fefb (*)
FCIP Tunnel traces (32-64):  On     Trigger Mask:  0x00000001 (T)
TCPIP traces (65):           On     Display Mask:  0x8000fefb (-)
TCPIP Conn. traces (66):     Off    Tunnel Mask:   Inactive
IP traces (67-83):           Off    Post trigger:  3% - 3600 events
ARL traces (84):             Off    Record Size:   128
ETHERNET traces (85-103):    Off    Auto Checkout: Enabled
IP API traces (104):         Off    FTRACE is:     Enabled
FCIP MSG traces (105):       Off    Debug level:   4-Normal (low)
VDM traces (106):            Off
 *-Bit 31 [0x80000000]: Software Structure
```

```
   Bit 19 [0x00080000]: EtRX - Ethernet Received Frame
   Bit 18 [0x00040000]: EtSX - Ethernet Send Frame to Peer
   Bit 17 [0x00020000]: TnTX - Tunnel Received Peer Frame
   Bit 16 [0x00010000]: TnSX - Tunnel Send Frame to Peer
 *-Bit 15 [0x00008000]: FcT - FC FWD Frame From Peer
 *-Bit 14 [0x00004000]: FcR - FC FWD Received Frame
 *-Bit 13 [0x00002000]: Dsc - Discarded Frame
 *-Bit 12 [0x00001000]: Data - Frame Data
 *-Bit 11 [0x00000800]: State Change
 *-Bit 10 [0x00000400]: CpRX - Frame Received From CP
 *-Bit  9 [0x00000200]: CpSX - Frame Sent To CP
   Bit  8 [0x00000100]: ToP - Sent To Peer
 *-Bit  7 [0x00000080]: Tfx - Emulation FC Frame From Peer
 *-Bit  6 [0x00000040]: Rfx - Emulation FC Received Frame
 *-Bit  5 [0x00000020]: Sfx - Send Frame
 *-Bit  4 [0x00000010]: Gfx - Generated Frame
 *-Bit  3 [0x00000008]: FC SOFi1/2/3 or Class F Frames
   Bit  2 [0x00000004]: FC SOFn1/2/3 Frames
 *-Bit  1 [0x00000002]: Msg - Information
T*-Bit  0 [0x00000001]: Err - Error
+-----+----------+--------+------------+-------+------+------+--------+--------+
|     |          |        |Trace Header| Wrap  | In   | Out  | Switch | Switch |
|  Id | State    |  Size  |  Address   | Count | OXID | OXID |  Date  |  Time  |
+-----+----------+--------+------------+-------+------+------+--------+--------+
|   1 | Current  | 100000 | 0x001f2f00 | 12344 | FFFF | FFFF |        |        |
|   1 |   unused | 100000 | 0x001f3180 |     0 | FFFF | FFFF |        |        |
|   2 |   unused | 100000 | 0x001f3400 |     0 | FFFF | FFFF |        |        |
|   3 |   unused | 100000 | 0x001f3680 |     0 | FFFF | FFFF |        |        |
+-----+----------+--------+------------+-------+------+------+--------+--------+
```

The table at the bottom of the output example has the following information:

- Id — The FTRACE trace buffer identifier or buffer number.
- State — The FTRACE buffer state for that buffer number. The state can be one of the following:
  - Current — The buffer is the current active buffer in use for FCIP events
  - Triggered — The buffer has been used to record an error event from the FCIP complex. This state is used only when the Auto Checkout option was disabled.
  - Checked Out — The buffer has been used to record an error event from the FCIP complex, and the buffer will not be overwritten.
  - Post Fill — A trigger event has been encountered, and the FTRACE buffer is currently being post-filled with a number of post-error events. Once the post-filling has been completed, the buffer will transition to either a “Checked Out” or “Triggered” state.
  - Unused — The buffer has not been used to capture any FCIP events. The buffer will be used when the prior buffer in the list transitions to either a “Checked Out” or “Triggered” state.
- Size — The number of trace records that are in the buffer.
- Trace Header Address — A memory address used internally for controlling access to the trace buffer.
- Wrap Count — The number of times that a trace buffer has been wrapped. The trace buffer is a circular buffer that wraps after the size number of trace events has been exceeded.
- In OXID and Out OXID — Not used until the buffer is being analyzed.
- Switch Date — Indicates the system date when the buffer transitioned to either a “Checked Out” or “Triggered” state.
- Switch Time — Indicates the system time when the buffer transitioned to either a “Checked Out” or “Triggered” state.

Index