BROCADE
A Broadcom Company

SAN Design and Best Practices

Brocade® Fibre Channel Platforms

Design Guide

Copyright © 2016–2026 Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, go to www.broadcom.com. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

The product described by this document may contain open source software covered by the GNU General Public License or other open source license agreements. To find out which open source software is included in Brocade products or to view the licensing terms applicable to the open source software, please download the open source attribution disclosure document in the Broadcom Support Portal. If you do not have a support account or are unable to log in, please contact your support provider for this information.

Table of Contents

Chapter 1: Preface 7
1.1 Introduction 7
1.2 Audience and Scope 7
1.3 Approach 8
1.4 Overview 8
Chapter 2: Storage Landscape 9
2.1 The Storage Landscape 9
2.2 Tipping Point: The All-Flash Data Center 9
2.3 NVMe 11
Chapter 3: Architecting a SAN 13
3.1 Operational Considerations 14
3.2 Be the Pilot 14
3.3 Predeployment Cabling and Optics Validation 14
Chapter 4: SAN Design Basics 16
4.1 Topologies 16
4.1.1 Collapsed-Core Topology 17
4.1.2 Core-Edge Topology 17
4.1.3 Full-Mesh Topology 17
4.2 High-Performance Latency-Sensitive Workloads 17
4.3 Redundancy and Resiliency 18
4.4 Switch Interconnections 19
4.4.1 Brocade UltraScale ICL Connectivity 20
4.5 Brocade UltraScale ICL (Inter-Chassis Link) Best Practices 21
4.6 Full-Mesh Topology 21
4.7 Device Placement 23
4.7.1 Traffic Locality 23
Chapter 5: Data Flow Considerations 25
5.1 Fan-In Ratios and Oversubscription 25
Chapter 6: Scalability and Performance 26
Chapter 7: Supportability 28
7.1 Firmware Upgrade Considerations 29
Chapter 8: Monitoring 30
8.1 Brocade Fabric Vision Technology 30
8.1.1 MAPS 30
8.1.1.1 MAPS Recommendations 30
8.1.1.2 Tips on Getting Started with MAPS 31

8.1.2 Fabric Performance Impact Monitoring 31
8.1.3 SDDQ Explained 31
8.1.4 Flow Vision 32
8.1.5 IO Insight 32
8.1.6 VM Insight 32
8.2 SANnav Management Portal Monitoring Overview 32
8.3 Troubleshooting 33
8.3.1 ClearLink (D_Port) 33
8.3.2 Recommendation: D_Port On-Demand 33
8.3.3 FEC 33
8.3.4 Buffer Credit Loss Detection and Recovery 34
8.3.5 RASLog Messages 34
8.3.6 Audit Log Messages 34
8.4 Monitoring the Switches 35
8.5 Latencies 35
8.6 Misbehaving Devices 35
8.7 Design Guidelines 36
Chapter 9: Fibre Channel Routing 37
9.1 Overview and Purpose 37
9.2 Edge Fabrics 37
9.3 Inter-Fabric Links 37
9.4 Backbone Fabrics 37
9.5 Redundancy 42
9.6 Avoiding Congestion 42
9.7 Available Paths 43
9.8 FCR and Extension 43
9.9 FCR Design Guidelines and Constraints 43
Chapter 10: Virtual Fabrics 45
10.1 Use Case: FICON and Open Systems (Intermix) 45
Chapter 11: Intelligent Fabric Services 46
11.1 In-flight Encryption and Compression 46
11.1.1 Virtual Fabric Considerations: Encryption and Compression 46
11.2 Fabric Notifications 46
11.3 Adaptive Traffic Optimizer 47
11.3.1 Adaptive Traffic Optimizer 48
Chapter 12: Extension 49
Chapter 13: SAN Design for Critical Workloads 50
13.1 Placement of Servers with Business-Critical Workloads 50
13.2 Business-Critical VMs 50

Chapter 14: Access Gateway and NPIV 51
14.1 Benefits of Brocade Access Gateway 52
14.2 Constraints 53
14.3 Design Guidelines 53
14.4 Monitoring 53
14.5 Maintenance 54
14.6 Access Gateway Mapping 54
14.6.1 Port Mapping 54
14.6.2 Device Mapping 54
14.6.3 Default Port Mapping 55
Chapter 15: Security 57
15.1 Zoning: Controlling Device Communication 57
15.1.1 Peer Zoning 57
15.1.2 Target-Driven Zoning 58
15.1.3 Zone Management: Duplicate WWNs 58
15.2 Securing the SAN Infrastructure 59
15.3 Access Control Lists (ACLs) 60
15.3.1 Switch Connection Control (SCC) Policy 60
15.3.2 Fabric Configuration Server (FCS) Policy 60
15.3.3 Device Connection Control (DCC) Policy 60
15.3.4 Policy Database Distribution 60
15.3.5 Authentication Protocols 61
15.4 Principle of Least Privilege (PoLP) 61
15.5 Secure SAN Management 61
15.5.1 Role-Based Access Control (RBAC) 61
15.6 Securing Management Interfaces 62
15.6.1 IP Filter 62
Chapter 16: Automation 63
16.1 Overview and Purpose 63
16.2 Motivation to Automate 63
16.3 Overview of the REST API 64
16.4 Simple Automation Example 65
16.5 Ansible as an Alternative 67
16.6 SANnav REST API 68
16.7 Conclusion 68
Appendix A: Optical Cables 69
Appendix B: Fabric Details 70
Appendix C: References 74
C.1 Compatibility, Scalability, and Target Path 74

C.2 Brocade SAN Health 74
C.3 Brocade Bookshelf 74
C.4 Other 74
Revision History 75
53-1004781-08; November 18, 2025 75
53-1004781-07; November 18, 2025 75
53-1004781-06; May 10, 2024 75
53-1004781-05; August 15, 2023 75
53-1004781-04; July 2023 75
53-1004781-03; January 2023 75
53-1004781-03; May 2022 75
53-1004781-02; September 1, 2020 75
53-1004781-01; November 23, 2016 75

Chapter 1: Preface

1.1 Introduction

The storage landscape continues to modernize, and we must make informed decisions to design an exemplary Fibre Channel architecture. This document serves as a high-level storage area networking (SAN) design and best practices guide for Brocade® products and features, primarily focusing on Fibre Channel SAN design and implementation. Topics include the early planning phase, understanding possible operational challenges, and monitoring and improving an existing SAN infrastructure.

The guidelines in this document do not apply to every environment, but they will help you make decisions for a successful SAN design. For details about the hardware and software products, contact your Broadcom representative or refer to the documents in Appendix D.

NOTE: This is a living document that is updated frequently. Check www.broadcom.com for the latest document version and other best practice documents.

1.2 Audience and Scope

This guide is for IT architects who are directly or indirectly responsible for SAN design based on Brocade Fibre Channel platforms. It describes the challenges faced by SAN designers in greenfield and legacy environments. While not intended as a definitive design document, this guide introduces concepts and guidelines to help avoid potential issues resulting from poor design.

This document describes the best practice guidelines in the following areas:

- Modernizing the storage landscape
- Architecting a SAN
- SAN topologies
- Data flows
- Adaptive Traffic Optimizer
- Fabric performance impact notification (FPIN)
- Predeployment infrastructure testing
- Device connections
- Scalability and performance
- Supportability
- Monitoring
- Troubleshooting
- Fibre Channel routing (FCR)
- Intelligent services
- NPIV
- Access Gateway
- Workloads
- SAN management
- Security
- Automation

NOTE: A solid understanding of SAN concepts and Brocade Fibre Channel technology is assumed. See Appendix C for recommended other publications.

1.3 Approach

Although some advanced features and specialized SAN applications are discussed, these topics are covered in greater detail in separate documents. The primary objective of this guide is to provide a solid foundation to facilitate successful SAN designs. Successful designs effectively meet current and future requirements. This document addresses basic administration and maintenance, including capabilities to identify early warning signs for end-device (initiator or target) latency, which can cause congestion in the SAN fabric. Comprehensive discussions of SAN fabric administration, storage network cabling, and Fibre Channel security best practices are covered in separate documents.

1.4 Overview

Although Brocade SAN fabrics are plug-and-play and can properly function if left in a default state, Fibre Channel networks benefit from a well-thought-out design and deployment strategy. Your SAN topology should adhere to best practice guidelines for reliable and efficient data delivery. The best practice guidelines in this guide are based on SAN industry standards, tremendous experience, and considerations specific to Broadcom® products.

This document does not consider physical environment factors such as power, cooling, and rack layout. Instead, this document focuses on network connectivity, edge devices to the fabric, inter-switch links (ISLs), and software configurations.

NOTE: This document has a switch-centric scope, and it does not discuss HBA, storage, end-device setup, configuration, or maintenance. It does cover some aspects of fabric monitoring, management, diagnostics, cabling, and migrations, but refer to other relevant documents for detailed information.

Chapter 2: Storage Landscape

2.1 The Storage Landscape

In the IT infrastructure world, storage is critical; it is where data resides, secure copies are stored, and application performance is founded. No matter how many CPU cores or how much memory a server might have, every server waits for data.

Consequently, the scope of this environment includes early disk drives, tape for securely and cost-effectively backing up the data, and software implementations that provide access, performance, and security. Furthermore, the storage administrator's responsibilities include securing copies of the data through RAID configurations or site-to-site replication solutions. The mantra of storage admins is, “a single copy of any data set is a single point of failure waiting for a disaster to occur”. Data loss is never an acceptable option from the application point of view.

2.2 Tipping Point: The All-Flash Data Center

Over the years, the state of the storage environments in IT has undergone significant changes. A brief review of the changes takes you from the early tape systems through the evolution of the hard disk drive (HDD) into the development of RAID systems and enterprise arrays. One of the things that had traditionally been true was that storage, based on HDD building blocks, evolved slowly. Changing from 5400 RPM disk drives to 7200 RPM disk drives as a performance enhancement (more data under the head per second) took 10 years to fully populate data centers. Other developments included the density of the magnetic signature on the drive platter and the number of platters and heads per drive. This retrospective is only helpful by denoting that the storage environment in IT did not progress as rapidly as, say, CPU development or memory performance and capacity.

Those developments were in silicon, and the drive development was mechanical. As a result, Moore’s Law applied to CPU and memory but not to storage. The nature of storage has undergone a significant shift with the advent of solid-state drives (SSDs).

Initially, the progress was moderate. In a brilliant market-enabling move, the drive vendors made the SSD platform the same shape and size as existing HDDs with the same SCSI, SAS, and SATA connectors. The result of this change meant plug compatibility on the back end of the array for the new technology. However, early on the enterprise array controllers, and in the case of embedded disk drives in servers, the OS driver stack did not take advantage of the improvements in performance and other drive characteristics. The OS stack not taking advantage of the new technology was part of why the early hybrid arrays, which used a mix of traditional HDD and SSD drives on the back end, were less performant than many storage administrators had expected. Not meeting expectations impacted the adoption rate, and the IT organization did not experience as much of an issue with existing storage networks as expected. For over 40 years in IT we have been addressing bottlenecks in CPU performance, memory speed/scale, storage capacity, performance, and network speeds. Removing one bottleneck lets you find the next one, not unlike how widening one segment of a major highway pushes the traffic backup to the next narrow section of the highway. Many customers felt that the performance gain versus the technology cost was only fit for their very highest-demand applications.

The cost per terabyte became more enticing with the advent of the all-flash array environment and inline features, such as compression, encryption, and deduplication. Additionally, the newer enterprise array controllers were designed for all-flash performance characteristics and significantly increased both input/output operations per second (IOPS) and latency. Added to that change was the storage density, which allowed collapsing multiple racks of HDD platforms into a partial rack of SSDs with the benefits of power and cooling reduction. What was the consequence? All-flash arrays have a significantly faster adoption rate, accounting for more than 70% of the shipping environment.

That technology shift leads to changing demands on the design of storage area networks, which is true regardless of the technology used. If the expectation is to utilize the capacity and performance of these platforms, then serious consideration must be given to the design.

A dedicated storage network infrastructure that provides lossless, low-latency, deterministic, scalable, and performant storage services to the applications becomes critical. Storage administrators will talk about fan-in or fan-out ratios for storage platforms, which are the number of servers and applications in the network using a particular array or array port for their storage access.

Depending upon the types of applications and their performance needs, that ratio might range anywhere from low single digits to 40 to 50 servers (hundreds of virtual machines). As with any provisioning scenario, the storage administrator deals with projections of how much capacity and performance any server or application will use. But application performance is variable. The time of day, week, month, and season, or event-driven events can cause spikes or drops in application demand.

Another consideration is that the entire application base does not refresh simultaneously. The typical scenario is that multiple generations of performance will simultaneously exist in the environment. The availability of service windows drives the need to re-platform existing environments to new servers and storage. Some legacy applications might not have an environment that runs a current operating system or application version. You might have a 10 year old or older operating system connected with an HBA with two or more older generations of technology behind it. How do you balance that still critical application against the needs and performance of the newer machines?

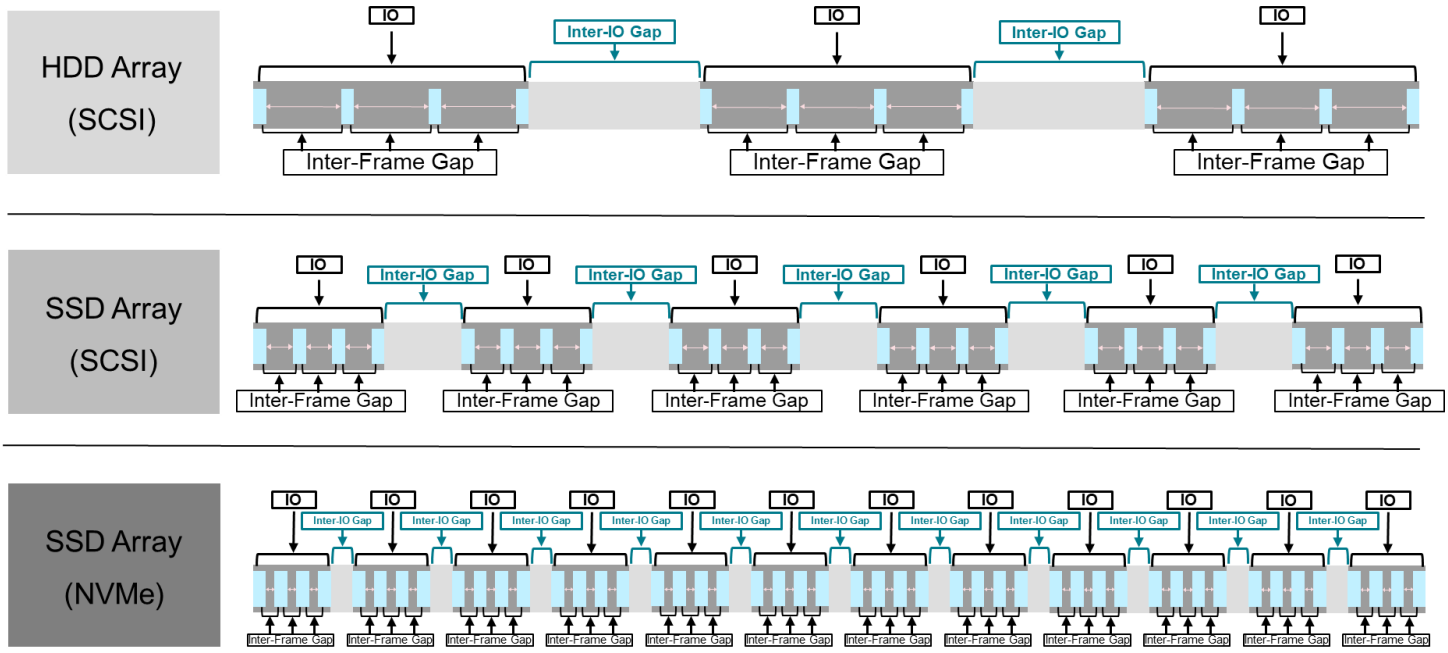
The answer combines topology, balanced provisioning, granular monitoring, and automated mitigation.

The most flexible configuration is a core-edge topology. Such an architecture enables significant scaling while maintaining a low number of hops, reducing the number of times the data transfers from one platform to another.

From a performance perspective, it is essential to note that the advent of the all-flash data center also means that new storage technologies, performance-based, capacity-based, or both, arrive at 18- to 24-month intervals. Technology refresh does not imply the wholesale replacement of the existing platforms, but rather that your storage network must be able to accommodate roughly two of these iterations per 4- to 5-year capital depreciation cycle. One advantage of a Fibre Channel SAN is the dual-redundant hardware and isolated nature of SAN architectures. A and B fabric architectures ensure that no device defect, scheduled maintenance, accidental human event, or malicious activity will completely take storage connectivity offline. Keeping storage networking online allows for seamless technology refreshes, which includes upgrades for storage network elements, attached servers, and storage; particularly in comparison to any hyper-converged architecture.

One of the additional changes required by the all-flash data center is improved monitoring, partly due to the reduction in latency, the criticality of data, and the high amount of data in-flight in modern SANs.

Figure 1: NVMe Implies Less Idle Time on the Network



As shown in the previous figure, the amount of idle time in the network continues to decrease. Consequently, the event window of a problem can be brief. Traditional monitoring systems, which sample flows, inspecting one frame in 8000, may be insufficient and cannot provide rapid root-cause analysis. The scale of modern all-flash storage further exacerbates the situation. Storage platforms can scale to over a petabyte (PB), 1000 terabytes (TB) of capacity, within two rack units. The consideration is that such platforms could host between 6000 and 10,000 virtual machines or applications. Any problem or outage affecting that kind of footprint becomes intolerable. Within the SAN, high availability requires granular monitoring with self-optimizing and self-healing technology. Humans are not fast enough or responsive enough to address problems in an all-flash data center before they occur; requiring reactive responses to an issue.

2.3 NVMe

While NVMe has become the dominant protocol for high-performance storage, particularly in SSDs and enterprise storage, it has not entirely replaced SCSI. SCSI, primarily through Serial Attached SCSI (SAS), still plays a role, especially in enterprise hardware and storage area networks. Another consideration why storage networks must be reconsidered and re-architected is non-volatile memory express (NVMe). At the device level, there are some characteristics to be aware of:

- Density – Current NVMe devices have 8 to 10 times the density of DRAM.
- Latency – Current NVMe devices have a sub-20-microsecond latency.
- Bandwidth – Current NVMe PCIe is a dual-simplex standard, it can transfer data in both directions simultaneously. An x16 slot, for example, provides 64 GB/s in each direction, for a total bidirectional bandwidth of 128 GB/s.
- Streamlined software – Current NVMe software has 13 required commands and 25 optional commands.

Storage density continues to increase on a rough Moore's Law schedule. Device latency continues to decrease, and it is significantly lower than HDDs. Network environments below 64G Fibre Channel are potential choke points for fan-in and fan-out ratios supported by large-scale storage platforms. SCSI, the legacy language used to communicate with storage, is changing for the first time in over four decades. Taking advantage of newer technology, especially its performance aspects, requires special attention to the storage network. Legacy environments, whether Ethernet or Fibre Channel, cannot take full advantage of NVMe performance. Reducing application overhead with streamlined software stacks, scaling IOPS, and driving consolidation depend on infrastructure. An NVMe over Fibre Channel fabric is a production-ready environment for NMVe.

Conversion will be slow. It will take time for all servers and applications to migrate from SCSI to NVMe. With the proper design and implementation of a Fibre Channel SAN, NVMe and SCSI can run concurrently on the same HBA, Fibre Channel switch, and storage. Potentially, applications can migrate nondisruptively from SCSI to NVMe.

However, this new environment must be self-learning, self-optimizing, and self-healing simply because it will be too critical and performant to wait for human intervention to solve problems before they become disruptive.

Chapter 3: Architecting a SAN

The SAN planning process is similar to any project planning process, and it includes the following phases:

- Phase I: Gathering requirements
- Phase II: Developing technical specifications
- Phase III: Estimating project costs
- Phase IV: Analyzing return on investment (ROI) or total cost of ownership (TCO) (if necessary)
- Phase V: Creating a detailed SAN design and implementation plan

When selecting which criteria to meet, you should engage users, server and storage subject matter experts (SMEs), and other relevant experts to understand the role of the fabric. Since most SANs operate for a long time before they are renewed, consider future growth, as SANs are complex to re-architect. Deploying new SANs or expanding existing ones to meet additional workloads in the fabrics requires a critical assessment of business and technology requirements. Proper focus on planning will ensure that the SAN, once deployed, meets all current and future business objectives. These objectives include availability, deployment simplicity, performance, future business growth, and cost efficiency. Tables in Appendix B are provided as a reference for documenting assets and metrics for SAN projects.

A critical aspect of a successful implementation that is often overlooked is the ongoing management of the fabric. Identifying SMEs for all components that make up the SAN, and adequate and up-to-date training on those components is critical for efficient design and operational management of the fabric.

When designing a new SAN or expanding an existing SAN, you should consider the following parameters:

- **Application Virtualization**
  - Which applications will run in a VM environment?
    - How many VMs will run on a physical server?
    - Under what conditions will the VMs be migrated (business and non-business hours; is additional CPU or memory needed to maintain response times)?
  - Solid-state storage and NVMe response times?
- **Homogeneous/Heterogeneous Server and Storage Platforms**
  - Are blade servers or rack servers used?
  - Is auto-tiering in place?
  - Which Brocade Fabric OS® versions are supported in a multivendor storage environment?
  - What is the planned refresh cycle of servers and storage platforms (2 or 3 years)?

Scalability

  - How many server and storage ports are needed now?
    - How many devices will connect through an Access Gateway?
    - How many ISLs and Brocade UltraScale inter-chassis links (ICLs) are required to minimize congestion in the fabric?
    - What distances for ISL and ICL connections need to be supported?
    - Does the fabric scale out at the edge of the core?
- **Backup and Disaster Tolerance**
  - Is there a centralized backup? (This determines the number of ISLs needed to minimize congestion at peak loads.)
  - What is the impact of backup on latency-sensitive applications?
  - Is the disaster solution based on long-distance metro FC ISLs or a Fibre Channel over Internet Protocol (FCIP) solution?

- **Diagnostics and Manageability**
  - What is the primary management interface to the SAN (CLI, REST, Brocade SANnav™ Management Portal, or third-party tool)?
    - How often will the Brocade FOS and SANnav Management Portal software be updated?
    - How is cable and optics integrity validated?
- **Investment Protection**
  - Is support needed for adding newer Gen 8 platforms into an existing older fabric?
  - Is support needed for storage technologies like NVMe over fabrics?
  - What device interoperability support is required?
  - Is interoperability required for other technologies such as UCS?

3.1 Operational Considerations

Even though Brocade fabrics scale in port density and performance, the design goal should prioritize simplicity, high level of availability, future expansion, and serviceability. Examples of this simplicity include using a two-tier core-edge topology, avoiding FCR, Virtual Fabrics when not required, and enabling port monitoring parameters for critical applications.

NOTE: Refer to the *Brocade SAN Scalability Guidelines* for currently tested and supported scalability limits. Any requirements that exceed the tested scalability limits should be pretested in a non-production environment to ensure they meet the necessary standards. Additionally, monitor system resources, such as CPU and memory utilization, to minimize fabric anomalies.

3.2 Be the Pilot

Whether building a new SAN or connecting to an existing one, validating the fabrics and prestaging the applications before production ensures baseline metrics for expected throughput, latency, and errors based on the physical cable infrastructure.

3.3 Predeployment Cabling and Optics Validation

Brocade platforms are equipped with ClearLink™ Diagnostics. ClearLink Diagnostics enables predeployment testing to validate the integrity of the physical network infrastructure before operational deployment. Part of Brocade Fabric Vision®, a ClearLink Diagnostic Port (D_Port) converts a Fibre Channel production port into a diagnostic port for testing components and traffic. The test results can be beneficial in diagnosing a variety of port and link problems. ClearLink Diagnostics is an offline diagnostics tool that allows users to perform an automated battery of tests to measure and validate maximum speeds, latency, and distance across fiber infrastructure. ClearLink Diagnostics can verify the health and integrity of transceivers. Before deployment, users should conduct diagnostics to identify and address potential CRC errors caused by physical-layer issues, such as dirty optics, faulty cables, and damaged connectors.

D_Port requires the production port being tested, to be offline. All other production ports are unaffected. A D_Port can test links to a new fabric switch without allowing the new switch to join the fabric.

ClearLink Diagnostics is a fabric-based, physical-layer validation that enables the following metrics:

- Transceiver health check
- Transceiver uptime
- Long-distance measurements – Link distance is reported for links 1 KM or longer
- Link latency measurements between D_Ports
- Link power loss (dB)
- Link performance

Refer to the *Brocade Fabric OS Troubleshooting and Diagnostics User Guide* for a more detailed discussion of D_Port usage.

Refer to “Appendix A: ClearLink Diagnostics” in the *SAN Fabric Resiliency and Administration Best Practices User Guide* for details about enhancements in each FOS release.

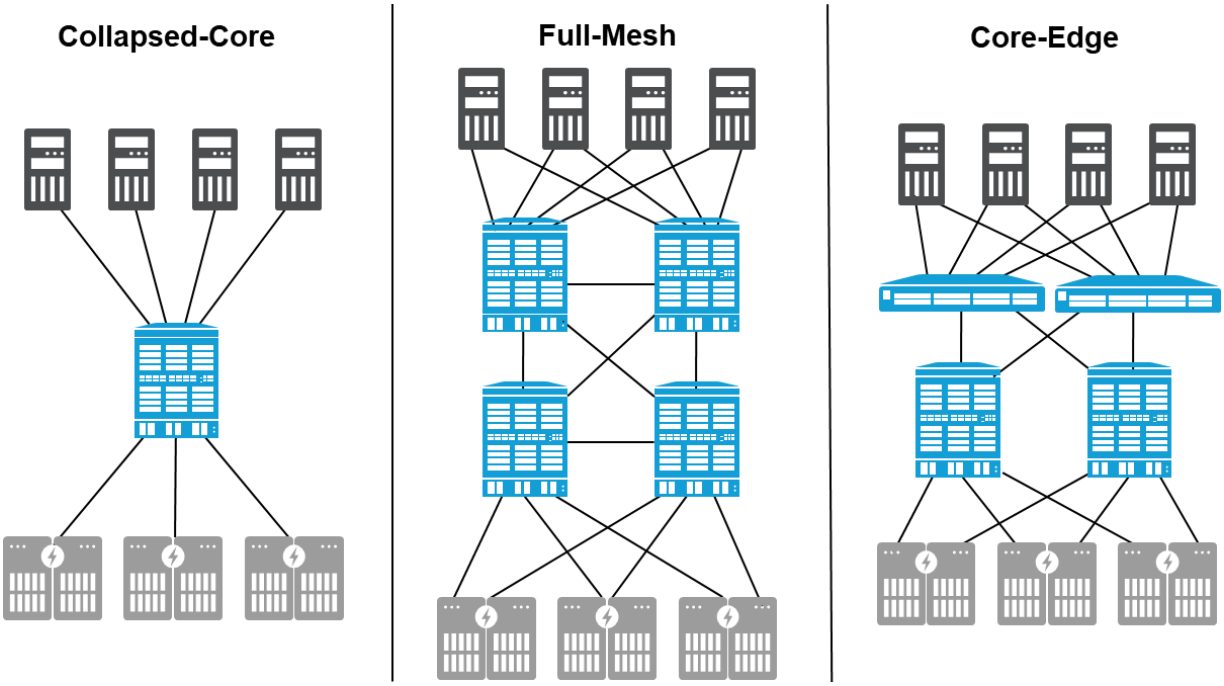
Chapter 4: SAN Design Basics

This chapter provides high-level guidelines for architecting a typical SAN. It focuses on best practices for collapsed-core, core-edge, and mesh fabrics. The discussion starts at the highest level, the data center, and works down to the port level, providing recommendations at each point along the way.

4.1 Topologies

A typical SAN architecture comprises edge devices, network devices, and cabling. Topology is usually described in terms of interconnected switches, such as collapsed-core, core-edge, and full-mesh. The recommended SAN topology to optimize performance, availability, management, and scalability is a tiered, core-edge topology. The core-edge approach provides excellent performance without unnecessary interconnections. At a high level, the tiered topology utilizes many edge switches used for device connectivity and a smaller number of core switches used for routing traffic between the edge switches; see the following figure.

Figure 2: Three Scenarios of Tiered Network Topologies



The primary difference between these three scenarios is device placement, where devices are connected to the network and the associated traffic flows:

- Scenario A: A collapsed-core architecture localizes traffic to a single platform. On a platform, each ASIC performs local switching. The receiving ASIC either switches a flow out to the end device or sends it to the next closer ASIC. A collapsed-core can have small performance advantages for performance-critical, latency-sensitive workloads. Local switching does not scale beyond an ASIC's port group. A collapse-core architecture significantly reduces manageability by having one platform per fabric. The overall number of fabrics determines manageability.; most SANs have two fabrics.

- Scenario B: A core-edge separates storage and server connectivity, thus providing ease of management and greater scalability. A core-edge topology has only one fabric hop from server to storage, providing identical performance as full-mesh while allowing greater scalability. Core-edge SAN architecture is the most popularly deployed.
- Scenario C: A full-mesh fabric has no more than one hop between server and storage, assuming the server and storage are not connected to the same platform. Designing fabrics with UltraScale ICLs is an efficient way to save valuable FC ports. Using best-practice SAN design considerations, users can quickly build a large fabric with 3456 ports or more.

4.1.1 Collapsed-Core Topology

Collapsed-core topology is often employed for high-performance storage connectivity, particularly critical applications. In a collapsed-core topology, storage and servers connect to a single ASIC on a switch or director, which enables local switching. Redundant A and B fabrics are still utilized. A collapsed-core topology is a consideration that drives some administrators to place storage ports in the same port group as the server ports, whether on a blade or a switch.

The collapsed-core topology (Figure 2) places initiators (servers) and storage (targets) on the same chassis and potentially the same blade and ASIC. This topology has several benefits depending on the size of the environment. Collapsed-core topology is used when customers migrate from multiple switches to a single, dual-core architecture where all initiators and targets fit. Moving to a core-edge architecture is best if future design requirements include increasing capacity.

4.1.2 Core-Edge Topology

The core-edge topology (Figure 2) places initiators (servers) on the edge tier and targets (storage) on the core tier. For redundancy, each fabric (A and B) has two cores. Since servers and storage are on different switches, this topology provides easy management, outstanding performance, and minimal latency. The data flows traverse one hop from edge to core. Storage-to-storage traffic will require two hops if the second storage platform destination is not connected to the same core. The two cores within the same fabric can be connected if storage-to-storage connectivity is required. The disadvantage to a core-edge design is that storage and core-to-edge connections contend for expansion as the environment scales; however, director platforms are flexible, allowing ICLs for inter-switch connectivity and freeing up ports for additional devices.

4.1.3 Full-Mesh Topology

Another alternative topology is a full-mesh architecture. Every switch has direct ISLs to every other switch in the fabric. A full-mesh topology is problematic for environments with many switches, because ISLs consume valuable ports that could otherwise be used to connect servers and storage. Director platforms have ICLs, allowing exceptional bandwidth and scale between chassis without consuming ports. A full-mesh topology ensures that no end device is more than one hop away from another.

A full-mesh topology (Figure 2) allows you to place servers and storage anywhere since communication between source and destination is no more than one hop. This design utilizes director-class switches with UltraScale ICL ports for interconnectivity to ensure maximum device port availability and utilization. Design this architecture with a minimum of two switches and up to nine in a full-mesh topology.

4.2 High-Performance Latency-Sensitive Workloads

In the last decade, enterprises have come to leverage low-latency, high-throughput flash arrays for demanding, performance-sensitive workloads. Brocade Gen 8 Fibre Channel is ideally suited for these workloads due to the sub-microsecond latency through the switch and the high-speed bandwidth while providing accurate I/O instrumentation. Performance testing has demonstrated that all-flash arrays significantly benefit Gen 8 SANs.

Fibre Channel standards include forward error correction (FEC), RS(544,514) to ensure transmission reliability and highly deterministic data flows. A single FEC can correct up to fifteen 10-bit symbol errors and detect up to thirty 10-bit symbol errors. Frame retransmits are avoided when bit errors are corrected. For Gen 8, dual FEC is enabled per default.

For highly demanding workloads, a no-hop fabric connection through a one-ASIC switch, such as the Brocade G820, or local switching within an ASIC on a director's port blade minimizes latency to sub-microsecond speeds (measured in nanoseconds). Local switching performs cut-through switching of Fibre Channel frames from the ingress port to the egress port when both ports are in the same port group. Some platforms, such as the Brocade X7/X8 directors and G730 switches, traverse up to three switching ASICs between data ingress and egress. Keeping host and storage connections within an ASIC's port group, slightly minimizing latency by a couple hundred nanoseconds.

For details about port groups and local switching, refer to the *Brocade Fabric OS Administration Guide* and the hardware installation guide for the appropriate product.

4.3 Redundancy and Resiliency

An essential aspect of SAN architecture is fabric resiliency and redundancy. The objective is to remove a single point of failure. Resiliency is the network's ability to continue functioning after a failure. Redundancy describes the duplication of components, typically the entire fabric, to eliminate a fabric failure as a single point of failure.

Brocade fabrics have resiliency built into FOS, which runs on all Brocade platforms. FOS can repair and overcome failures. For example, the fabric shortest path first (FSPF) routing algorithm computes a new path when a fabric topology changes, like when a link goes offline, assuming a second path exists when fabric resiliency is essential.

The key to high availability and enterprise-class availability is redundancy. Business continuance is provided through most foreseeable and unforeseeable events by eliminating an entire fabric as a single point of failure. At the highest level of fabric design, the complete fabric should be redundant, with two mirrored, entirely different fabrics that do not share any common SAN platforms.

Servers and storage devices should be connected to both fabrics (A and B), leveraging some form of multipath I/O (MPIO) so that data can flow across both fabrics seamlessly in an active/active or active/passive mode. MPIO ensures that an alternate path is available if the current path fails. Ideally, redundant fabrics are identical, but they should be based on the same switches to ensure consistency of performance and delivery. Often, A and B fabrics are in the same location. However, two (local and asynchronous) or three (local, synchronous, and asynchronous) locations are frequently used to provide disaster recovery (DR), and a separate replication SAN is constructed. Operations can continue after a disaster, either at the remote location or after data has been retrieved.

Regardless of the physical geography, there are two production fabrics for redundancy, and frequently two replication fabrics. The replication fabrics connect directly to the arrays' replication ports; they do not connect to the production fabrics.

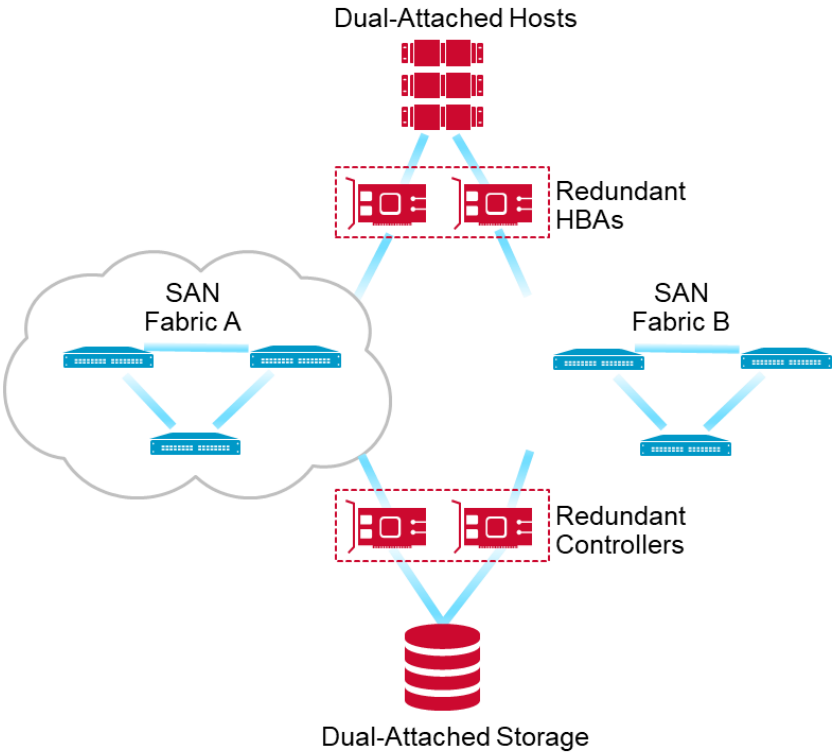
In summary, best practices for SAN design are to ensure application availability and resiliency through the following methods:

- Fabric redundancy to avoid a fabric being the single point of failure
- Resiliency is built into each fabric to prevent a single point of failure
- Redundant connections from each host to each fabric
- MPIO-based failover from initiator to target
- Identical architectures and platforms in each fabric
- Redundant ISL, inter-fabric link (IFL), and ICL for inter-switch connectivity
- Separate storage (core tier) and server (edge tier) tiers for independent expansion
- Core switches of equal or higher performance compared to the edge switches

4.4 Switch Interconnections

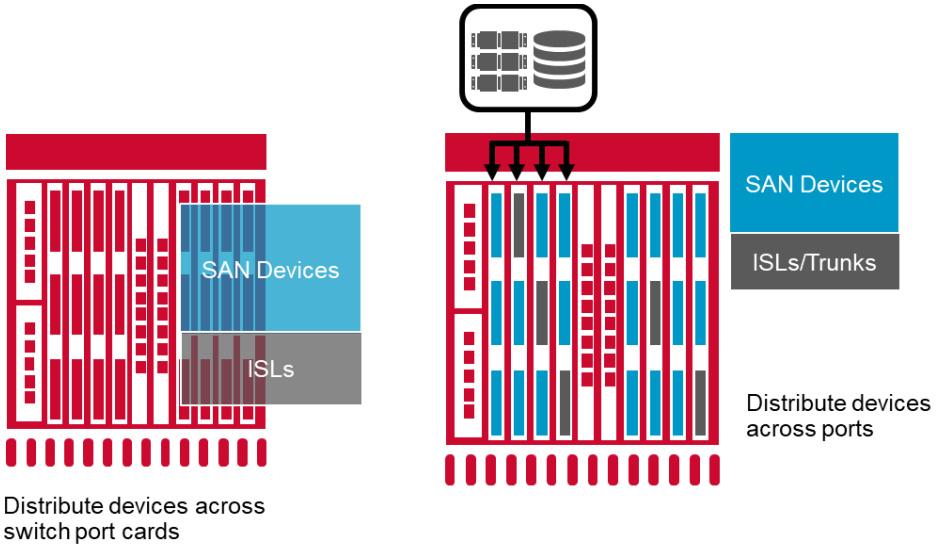
As mentioned previously, at least two of every element in the SAN should provide redundancy and improve resiliency. The number of available ports and device locality (server/storage tiered design) determines the number of ISLs needed to meet performance requirements. ISL requirements for directors include a minimum of two trunks with at least two ISLs per trunk. Each source switch should be connected to at least two other switches, and so on. In the following figure, each blue connection line represents at least two physical cables. Two physical connections provide redundancy for ports, optics, fiber patch cables, patch panels, and fiber infrastructure.

Figure 3: Connecting Devices through Redundant Fabrics



Redundant trunks on a director platform should be placed in varying port groups on different blades, as shown in Figure 4. See the appropriate hardware manual to determine port groups for the various models of port blades. For more details, refer to the *Brocade Fabric OS Administration Guide*. Whichever method is decided upon, it is crucial to be consistent across the SAN. For example, do not place ISLs on lower port numbers in one chassis, as shown on the left in Figure 4, and stagger ISLs on a different chassis, as shown on the right in Figure 4.

Figure 4: Examples of Distributed ISL (BT) Placement for Redundancy



NOTE: In the previous figure, bandwidth times (BTs) are placed on different port groups. Matching BT placement between devices and across fabrics is essential to ensure simplicity in design and assist in problem investigation.

4.4.1 Brocade UltraScale ICL Connectivity

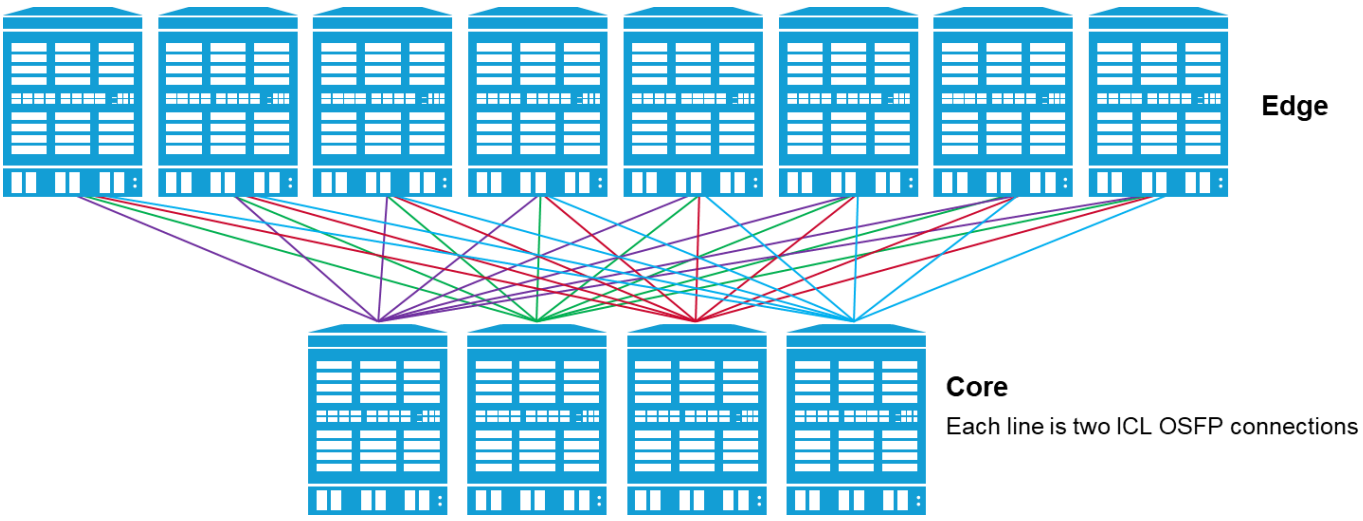
Brocade UltraScale ICL technology preserves Fibre Channel ports for end device connections. The Brocade X8-8 and X8-4 platforms use Brocade UltraScale ICL with optical OSFPs (Octal Small Form Pluggable). The Brocade X8-8 has two Brocade ICL8 X8-8 blades, each with eight OSFP ports, totaling sixteen OSFP ports per 8-slot director for ICL connectivity. In a Brocade X8-8 Director, each OSFP port has eight independent links providing a total of 128 ICL links. Each link connects to one of the eight core ASICs within the Brocade X8-8 Director.

The Brocade X8-4 has two Brocade ICL4 X8-4 blades, each with four OSPF ports, totaling eight OSFP ports per 4-slot director for ICL connectivity. In a Brocade X8-4 Director, each OSFP port has eight independent links providing a total of 64 ICL links. Of the eight links, two links connect to one of the four core ASICs within the X8-4 Director.

Both 100m and 2 km OSFP optics are available. ICL oversubscription with Gen 8 directors is the same as Gen 7 directors at 3.6:1, when the director is fully populated with 48-port blades.

The Brocade X7 uses Quad Small Form Pluggable (QSFP) optics. The Brocade X7-8 supports 32 QSFP ports per chassis (see the following figure), and the X7-4 supports 16 QSFP ports per chassis. Each ICL port has four independent links, each connect to a different ASIC.

Figure 5: UltraScale X8-8 ICL Core-Edge Architecture

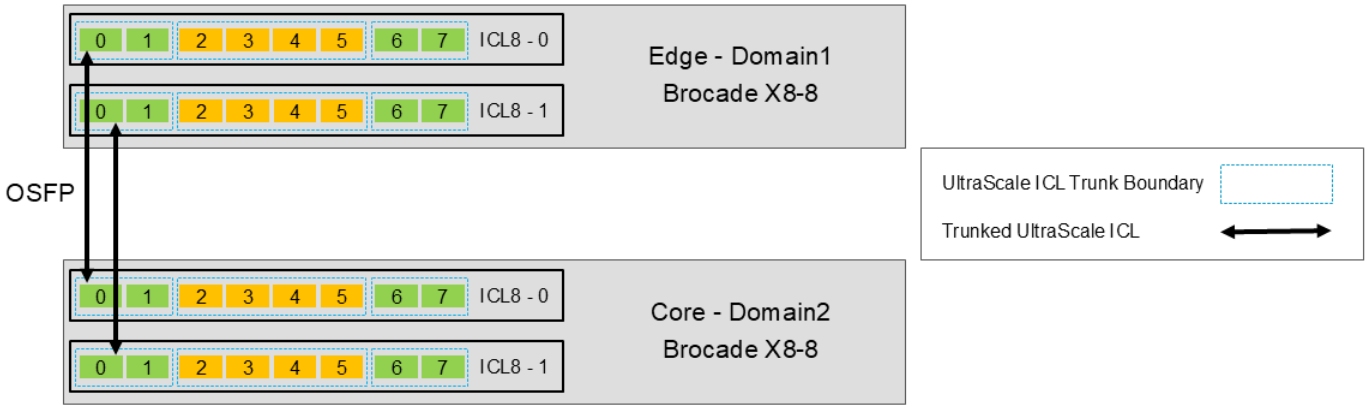


4.5 Brocade UltraScale ICL (Inter-Chassis Link) Best Practices

Each core ICL blade must be connected to an ICL blade in an edge chassis to achieve redundancy. See the following figure. In Figure 5, each X8-8 director has 16 ICL ports, eight per ICL blade. There are four core directors and eight edge directors. Each edge director ICL blade has a link to a core director ICL blade for a total of 16 ICLs. Eight edge directors × two ICL blades each, for a total of 16 ICLs per core director.

NOTE: A pair of ICL links is used to connect an ICL blade for redundancy.

Figure 6: Minimum ICL Connections between Brocade X8-8 Directors

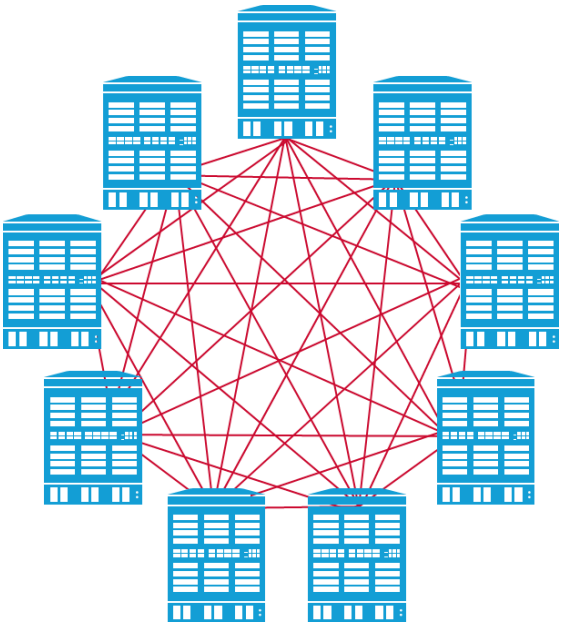


4.6 Full-Mesh Topology

A full-mesh architecture provides a single hop between source and destination. Broadcom supports a nine-director ICL mesh with up to 100 meter distances using QSFPs or OSFPs over OM4 fiber. In the example shown in the following figure, nine X8-8 directors with eight 64-port blades each, support 4608 128G Fibre Channel end-device ports. UltraScale ICLs with two ICL links between each director, equals 200 Gb/s between directors.

Alternatively, using UltraScale ICL connectivity, if the full-mesh had five X8-8 directors instead of nine, there would be 2560 128G Fibre Channel end-device ports with 400 Gb/s between each director.

Figure 7: Nine-Chassis UltraScale ICL-Based Full-Mesh Architecture



NOTE: Refer to the *Scale-Out Architecture with Brocade UltraScale Inter-Chassis Links Design Guide* for details. UltraScale ICL connections are considered a *hop of no concern* in a FICON fabric.

In a core-edge architecture, the edge switches should connect to two core switches using Brocade Trunks (BTs) with at least two ISLs. Each BT should be attached to a different blade. Redundancy requires a second mirrored fabric, and end devices must be connected to both fabrics using MPIO to manage active-active or active-passive flows and failover and failback flows.

The following recommendations are for ISL and UltraScale ICL connectivity:

- There should be at least two mirrored fabrics.
- There should be at least two core switches per fabric.
- Every edge switch should have at least two BTs to each core switch.
- Create small trunks. Keep BTs to two ISLs unless high traffic volumes are anticipated. Small trunks ensure that losing a BT does not result in significant bandwidth loss.
- Plan BTs based on one BT being offline.
- Place redundant BTs on different blades.
- BTs form only within a port group; port groups are in an ASIC boundary.
- Cable length difference within a BT should be identical for optimal performance.
- Use the exact cable length for UltraScale ICL connections.
- Use ISL or UltraScale ICL connectivity between the same domains; mixing the two types of connectivity is not supported.
  - Use the same optic type on both ends of an ISL:
  - Short-wavelength multi-mode optical fiber (SWL MMF)
  - Long-wavelength single-mode fiber (LWL SMF)
  - Extended-long-wavelength single-mode fiber (ELWL SMF)

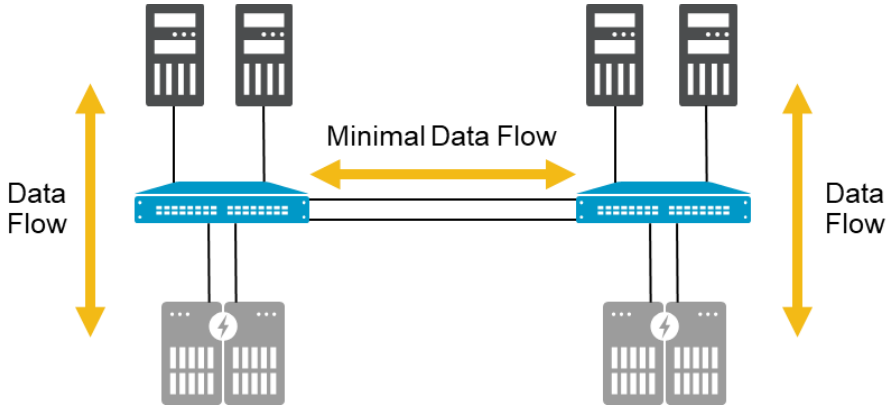
4.7 Device Placement

Device placement is a balance between traffic isolation, scalability, manageability, and serviceability. Virtualization has dramatically optimized compute platforms, driving the need for high performance and improved scalability in storage networks. Frame congestion can become a severe concern if there are end-device issues.

4.7.1 Traffic Locality

Designing device connectivity depends significantly on the expected data flow between devices. For simplicity, communicating hosts and targets can be attached to the same switch as shown in the following figure.

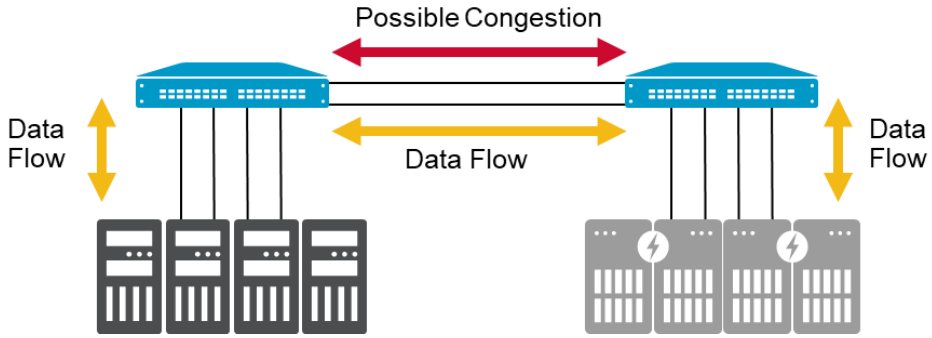
Figure 8: Hosts and Targets Attached to the Same Switch to Maximize Locality of Data Flow



This approach needs to scale. With Fibre Channel's high-speed, low-latency nature, attaching these host-target pairs on different switches does not mean performance is adversely impacted for typical workloads. With the current generation of switches, local switching is not required to gain performance or achieve low latency. Architects might want to switch traffic locally for mission-critical applications that depend on speedy response times.

In exceptional cases, multi-hop concerns might involve traffic congestion, specifically, inadequate inter-switch connectivity or concerns about proper resiliency as shown in the following figure. Often, these concerns can be mitigated by provisioning trunks/UltraScale ICLs.

Figure 9: Hosts and Targets Attached to Different Switches for Ease of Management and Expansion



A less common scheme for scaling a core-edge architecture is dividing the edge switches into a storage/target tier and a host/initiator tier. This approach is called edge-core-edge and lends itself to easier management and expansion. End devices do not connect to the core. Host and storage platforms have different performance requirements, cost structures, and other factors that are accommodated by placing initiators and targets in different tiers.

Chapter 5: Data Flow Considerations

5.1 Fan-In Ratios and Oversubscription

A critical aspect of data flow is the fan-in ratio, which is the oversubscription of initiator ports to target ports or edge devices to BTs. Alternatively, oversubscription can be viewed from the perspective of the storage array, referred to as the fan-out ratio. The ratio is the number of devices that share a single port, whether an ISL, BT, UltraScale ICL, or target port. Logically, a BT is a single port. The ratio is always expressed from the single entity's point of view, such as 7:1 for seven hosts utilizing a single BT or a single storage port.

What is the optimum number of hosts that should connect to a storage port? This question seems straightforward; however, the situation becomes complex once you consider clustered hosts, VMs, workload characteristics, and the number of LUNs per server. Determining how many hosts to connect to a particular storage port can be narrowed down to three considerations: port queue depth, IOPS, and throughput. Throughput is the only network component; thus, does a simple calculation adding up the expected peak bandwidth for each host suffice?

In practice, it is improbable that all hosts perform simultaneously at their maximum level. The bandwidth of the HBA was considerably overprovisioned with traditional application-per-server deployments. However, the game changed radically with the advent of VMs. Conceptually, oversubscription is built into virtual machines to optimize server resources. To the extent that servers optimize their resource utilization, they should proportionately increase their port utilization. Fewer virtual machine ports can oversubscribe a target port compared to non-virtualized machines. Nonetheless, it is prudent to oversubscribe ports to balance cost and performance. The difficult question is, by how much?

Another method is to assign host ports to storage ports based on the host's I/O capacity requirements. The intended result is a small number of high-capacity servers assigned to each storage port, effectively resulting in many low-capacity VM workloads distributed across the storage ports.

Port monitoring should determine the actual port utilization from fan-in and fan-out ratios, driving necessary adjustments. Ongoing monitoring provides valuable heuristic data for the successful expansion of storage and the efficient assignment of storage ports. A simple calculation determines the device-to-BT fan-in ratio.

Chapter 6: Scalability and Performance

Broadcom products are designed with scalability in mind, knowing that most installations will continue to expand and that growth is supported with few restrictions. However, by following the same basic principles outlined in the previous sections, performance and availability will continue to be ensured as the network grows.

Evaluate the impact on topology, data flow, workload, performance, and perhaps most importantly, redundancy and resiliency of the fabric when one of the following actions is performed:

- Add, move, change, or remove applications and traffic flows (databases)
  - Changes in workflows
  - Changes in provisioning
- Add, move, change, or remove initiators (hosts):
  - Changes in hardware (memory, processors, NIC, HBA, and so on)
  - Changes in virtualization
  - Changes in provisioning
- Add, move, change, or remove targets (storage):
  - Changes in virtualization
  - Changes in provisioning
  - Changes in storage media type; for example, HDD versus SSD
- Add, move, change, or remove fabric switches; for example, Gen 6 versus Gen 8
- Add, move, change, or remove ISL, IFL, ICL, and trunking (32G Fibre Channel versus 128G Fibre Channel)

If these best practices are followed when the fabric is deployed, small incremental changes should not adversely impact the availability and performance of the fabric. However, ongoing changes can negatively affect performance and availability if the fabric is not updated or adequately evaluated.

Some key points to cover when looking at the status of a production SAN include:

- **Reviewing redundancy and resiliency:**
  - Are there two or more redundant fabrics?
  - Are there two or more physically independent paths between each source (initiator) and destination (target) pair?
  - Does each host connect to two different edge switches?
  - Are edge switches connected to at least two different core switches?
  - Are inter-switch connections composed of two trunks of at least two ISLs?
  - Does each storage device connect to at least two different edge switches or separate port blades?
  - Are storage ports provisioned such that every host has at least two ports to access LUNs?
  - Are redundant power supplies attached to different power sources?
  - Are zoning and security policies configured to allow patch/device failover?
- **Reviewing performance requirements:**
  - Host-to-storage fan-in and fan-out ratios
  - Oversubscription ratios:
    - Host to ISL
    - Edge switch to core switch
    - Storage to ISL
  - Size of BTs
  - Routing policy and currently assigned routes (evaluate actual utilization for potential imbalances)

- **Investigating fabric latencies:**
  - Unexpectedly poor storage performance
  - Overloaded hosts or applications
  - Constrained long-distance links from usage changes, such as adding mirroring or workloads
  - Deteriorating optics resulting in declining signal strength and increased error rates

Storage response latency can be baselined and monitored continuously using IO Insight with Monitoring and Alerting Policy Suite (MAPS). Deal with latencies immediately; they can impact the fabric profoundly. In summary, although a Brocade SAN allows for any-to-any connectivity and supports provision-anywhere implementations, these practices can harm performance and availability if left unchecked. As detailed above, the network needs to be monitored for changes and routinely evaluated for meeting redundancy and resiliency requirements.

Chapter 7: Supportability

Supportability is a critical part of deploying a SAN. Follow the guidelines in this chapter to ensure that the data needed to diagnose fabric behavior or problems has been collected:

- Follow Brocade's management interface best practices for connecting to the data center LAN. Set up a different VLAN (broadcast domain) for the management interfaces belonging to each fabric and route between the VLANs. Do not put the management interfaces from fabric A in the same VLAN as the management interfaces from fabric B. This separation is part of the air gap between the fabrics for redundancy.
- Implement a serial console server. Implement remote serial access to fabric switches for times when there are network problems or issues involving a firmware upgrade or switch boot.
- Require every user to have an AAA account for logging so that individual user actions can be tracked. Configure individual user accounts on LDAP or RADIUS, and make personalized user accounts mandatory as part of a security policy.
- Restrict access by assigning or creating a pertinent role to each user.
- Enable auditing. Keep track of who made changes and when, only use individual user AAA accounts, and enable change tracking along with error logs and syslog forwarding.
- Enable Brocade MAPS. Leverage MAPS to proactively monitor errors and warnings such as CRC errors, loss of synchronization, and high-bandwidth utilization.
- Configure syslog forwarding. Troubleshooting can be expedited and simplified by sending log messages to a centralized syslog server. Forwarding messages to a syslog server is a simple monitoring functionality that historically maintains messages and expedites troubleshooting.
- Configure an NTP server. Configure switches to use an external time server to keep a consistent and correct date and time on all messages sent to a syslog server or other management devices.
- Create a switch configuration template in the SANnav Management Portal software to avoid configuration drift over time. You can adapt existing configurations as a template for quickly deploying new switches and ensuring consistency across the data center.
- Establish a testbed. Avoid missteps in a production environment. Set up a testbed to test: configuration changes, new applications, firmware updates, driver functionality, and scripts. Validate functionality and stability with rigorous testing in a test environment before deploying into the production environment.
- Use aliases to give switch ports and devices meaningful names for faster troubleshooting.
- Configure the `supportftp` command for automatic file transfers. The parameters set by this command are used by the `supportSave` and `traceDump` commands.

7.1 Firmware Upgrade Considerations

Both fixed-port and modular switches support hot code load (HCL) for firmware upgrades. A firmware update using HCL is nondisruptive:

- Disruptive versus nondisruptive upgrades:
  - During a nondisruptive upgrade, the fabric must remain stable without topology changes or added/removed devices
  - Simultaneous upgrades can be done on neighboring switches when the upgrade is nondisruptive.
  - Nondisruptive upgrades apply to standard Fibre Channel ports versus application and special-feature ports.
  - Extension offers nondisruptive upgrades for FCIP and IP Extension, as well as Fibre Channel ports.
- Review the *Brocade Fabric OS Release Notes* for the following information:
  - Upgrade path
  - Changes to feature support
  - Changes to backward compatibility
  - Known issues and defects

Consider a different Access Gateway firmware upgrade strategy. Brocade Access Gateways do not require the same firmware release level as Brocade switching platforms. Upgrading only directors and switches minimizes the infrastructure changes needed during an upgrade cycle.

Chapter 8: Monitoring

8.1 Brocade Fabric Vision Technology

Organizations need help managing data growth, protecting data, and leveraging actionable intelligence from data, all while meeting a strict SLA. As a result, even well-managed IT organizations must often make difficult choices about resource allocation, weighing the benefits of focusing more resources on monitoring and less on planning or optimizing. With Brocade Fabric Vision technology, organizations can achieve unprecedented insight and visibility across their critical storage network through advanced monitoring and diagnostic capabilities.

8.1.1 MAPS

MAPS is a health and threshold-based monitoring tool that allows for autonomous self-monitoring of directors and switches in the fabric. It helps detect potential and active problems, automatically alerting users to those problems long before they become costly outages. MAPS is a part of the Brocade Fabric Vision feature set.

MAPS tracks a variety of SAN fabric health categories and events. Monitoring fabric-wide events, ports, bit errors, and environmental parameters enable early fault detection and isolation as well as a means to measure performance. All health monitoring categories are customizable, providing flexibility around how and what users want to monitor. Create your own monitoring groups, assign custom thresholds, and with Fabric OS 9.0 and above, gain the same monitoring capabilities at a flow level. MAPS means users can now threshold-monitor application flows for abnormal completion times to manage SLAs.

Users can also easily integrate MAPS with enterprise operations solutions.

MAPS also provides predefined monitoring policies for users to get a quick start. These policies provide thresholds from 20 years of best practices and customer experiences. Users can select from conservative, moderate, or aggressive policies based on how closely users want to monitor their SAN environment. If the default policies do not meet your needs, customize the thresholds and actions, and activate your custom policy. MAPS provides notifications before problems arise, such as reporting overutilized ports approaching specified bandwidth limits, potentially leading to congestion. These insights enable SAN administrators to perform preemptive network maintenance, such as trunking or zoning, avoiding potential network failures.

MAPS lets you define how often switches and fabric elements are measured while specifying notification thresholds. Whenever fabric elements exceed these thresholds, MAPS can take action. These actions include administrative notifications using email, SNMPv3, RASlog, and automated actions, such as Slow Drain Device Quarantine (SDDQ), FPIN, and in some cases, port disable.

8.1.1.1 MAPS Recommendations

Brocade MAPS is a recommended feature that provides threshold monitoring of multiple switch elements. MAPS monitors ports based on type, each with unique monitoring and alerting thresholds. Different port types (F_Ports, E_Ports, VE_Port, and N_Ports) have different characteristics. MAPS provides flexibility in monitoring and alerting to address various cases.

MAPS allows for the monitoring and alerting of IO Insight flow metrics, which provides SAN administrators with notifications of performance degradation. An early alert of flow degradation could indicate congestion and associated response time impact. When support for VM Insight is enabled, it can identify potential end-to-end issues from an individual VM to its LUN.

8.1.1.2 Tips on Getting Started with MAPS

Are you new to the Brocade Monitoring and Alerting Policy Suite and want to start monitoring your SAN? MAPS provides deep insights into SAN health and performance with a single click. The following sections provide tips on the initial use of MAPS. A Fabric Vision license enables over 300 additional rules and is required to take advantage of all monitoring capabilities.

When starting with MAPS, SAN administrators should monitor their fabric with one of the three predefined policies (conservative, moderate, and aggressive). Start with the conservative policy to understand better what MAPS monitors, the severity of set thresholds, and the generated alerts. If the conservative policy does not meet your needs, switch to either the moderate or aggressive policy. SAN administrators can personalize default policies with their observed thresholds and desired actions to fit their environment better. SAN administrators can implement policy customization and management through the Brocade SANnav Management Portal software or the CLI.

The following items are some examples of customization:

- Clone predefined policies to customize individual thresholds and rules.
- Create custom monitoring groups, for example, ports, SFPs, and application flows.
- Distribute policies across the SAN for uniform fabric monitoring.
- Configure MAPS actions and take advantage of automated problem mitigation.
- Create custom MAPS monitoring dashboards through the SANnav Management Portal software.

8.1.2 Fabric Performance Impact Monitoring

Fabric Performance Impact (FPI) monitoring leverages predefined MAPS policies to automatically detect and alert administrators to the severity of latency and identify slow drain devices that could impact network performance. FPI detects various latency severity levels, pinpointing which devices are causing backpressure in the fabric and impacted by a bottlenecked port. MAPS and FPI work together to quarantine slow drain devices to prevent buffer credit starvation automatically.

8.1.3 SDDQ Explained

Lost buffer-to-buffer-credits (BBCs), credit-stall, and oversubscription lead to fabric congestion and backpressure. Backpressure potentially affects neighboring flows, referred to as victim flows, resulting in widespread performance degradation. The fabric uses automated SDDQ to mitigate backpressure using MAPS.

SDDQ works with MAPS and FPI monitoring to detect congestion scenarios and isolates problematic devices to a low-priority virtual channel (VC). By default, in a Brocade fabric, traffic runs on medium-priority VCs. Frame loss,

oversubscription, and impacted fabric performance cause SDDQ to automatically and non-disruptively quarantine egregious flows. The action MAPS takes is enabled individually for these conditions. Once problematic flows are isolated, fabric backpressure is relieved, which frees BBCs in the medium-priority VCs.

Also, FPI monitoring continually checks for cleared congestion conditions on impacted devices, allowing MAPS to automatically unquarantine previously quarantined flows. The MAPS unquarantine action moves flows back into the medium-priority VCs. This process is similar to the quarantine action and is non-disruptive. The MAPS SDDQ quarantine and unquarantine actions are supported on local and remote switches attached by ISLs and Brocade Access Gateways.

8.1.4 Flow Vision

Flow Vision is a diagnostics tool for Brocade SAN platforms. It provides traffic flow visibility in the fabric and can copy flows for later analysis. It allows test flow generation at line-rate speeds to prevalidate SAN hardware performance and connectivity. Use flow generation capability to confirm optimal health and assess your ability to support spikes in throughput.

For mission-critical applications, consider running Flow Vision constantly to keep a historical record of application performance profiles and identify intermittent irregularities. For application owners who might frequently call, run Flow Vision regularly when time permits to verify good fabric health and to preempt lurking issues.

8.1.5 IO Insight

IO Insight, also known as Flow Monitor, is supported by Brocade Fibre Channel platforms, which provide deeper flow-level IO statistics. These statistics include storage device latency and IOPS metrics such as first IO response time, IO completion time, and the number of pending IOs for a specific initiator and target or target and LUN. IO Insight provides IO workload monitoring and early detection of storage performance degradation.

It is a best practice to monitor critical applications with IO Insight. IO Insight metrics should be added to MAPS policies to better understand IO profiles receive notifications of storage IO performance degradation. This reporting is valuable for performance-sensitive workloads, enabling administrators to meet critical SLAs. IO Insight provides feedback on device reliability and performance optimization over time. The pending IO metric measures the HBA queue depth and can be used to fine-tune server queue depths.

Beginning with Fabric OS 9.0, IO Insight autonomously learns all flows traversing a switch with no user configuration required, as it is enabled by default. Once switches are discovered through the SANnav Management Portal, telemetry data is automatically propagated to the management platform, which is utilized for flow-level and application-level investigation.

Refer to the *Brocade Fabric OS Flow Vision User Guide* for configuration and usage details about Flow Vision and IO Insight.

8.1.6 VM Insight

The VM Insight feature provides the same IO and performance-level metrics that IO Insight provides for individual virtual machines. This feature distinguishes individual VM flows to the LUN, even if other VMs share the same LUN. VM Insight has unprecedented visibility and monitoring of VM application health and performance. It integrates with MAPS, allowing users to monitor thresholds and receive alerts on VM flow-performance deviations, similar to IO Insight.

This feature is available in Fabric OS 8.1 and later.

8.2 SANnav Management Portal Monitoring Overview

SANnav Management Portal is the Broadcom GUI-based management platform, tightly integrating with Fabric OS. From feature configuration to analysis of gathered telemetry data and events, the SANnav Management Portal software provides actionable insight to SAN administrators.

The Brocade SANnav Management Portal software uses Fabric OS features to detail device health, congestion, and flow telemetry, investigate concerns, troubleshoot issues, and customize dashboards. Refer to the SANnav Management Portal and Global View documents available here for more detail around the management platform monitoring capabilities.

8.3 Troubleshooting

8.3.1 ClearLink (D_Port)

A Brocade D_Port enables predeployment testing of the cable-plant infrastructure. Part of Brocade Fabric Vision technology, ClearLink is an offline diagnostics tool that allows users to perform an automated suite of tests to measure maximum throughput speeds, latency, and distance across links. ClearLink Diagnostics verifies the health and integrity of Fibre Channel transceivers in the fabric. The diagnostics are conducted before production or when excessive CRC errors occur.

A D_Port requires that the production port be taken offline. All other ports remain online and are isolated from D_Ports. ClearLink can also be used to test links to a new fabric switch without allowing the new switch to join or even be aware of the current fabric, providing an opportunity to measure and test ISLs before they are put into production.

ClearLink Diagnostics is a fabric-based, physical-layer validation that enables the following metrics:

- Transceiver health check
- Transceiver uptime
- Link power (dB) loss
- Link performance

8.3.2 Recommendation: D_Port On-Demand

When an on-demand D_Port-capable switch or chassis comes online, the switch checks if the other end of the connection supports dynamic D_Port mode. If dynamic D_Port is supported on the opposite end, the switch changes the remote port to D_Port mode and starts a diagnostic test automatically. After completing the test, the D_Port changes to normal port mode.

For Brocade ClearLink Diagnostics guidelines and restrictions, refer to the *Brocade Fabric OS Troubleshooting and Diagnostics User Guide* for a more detailed discussion of diagnostic port usage.

8.3.3 FEC

FEC is a data transmission error-correcting method that includes error-sensing data. Error-correcting code ensures virtually error-free transmission. FEC uses unused bits within the signaling protocol to generate an error-correcting code (ECC) and correct bits as needed. FEC supports the following data transmissions:

- When 10/16G FEC is enabled, it can correct one burst of up to 11-bit errors in every 2112-bit transmission, whether the error is in a frame or a primitive. Because FEC is optional at 10G and 16G speeds, the Transmitter Training Signal (TTS) was extended to negotiate FEC capabilities. FEC is negotiated and activated when both sides of the link have FEC enabled. The FEC active indicator in Fabric OS indicates whether FEC was successfully negotiated.
- 32G FEC is enabled by default as part of the Gen 6 standard, correcting up to seven symbols in every 5280-bit transmission. A symbol consists of 10 bits, so there are 528 symbols in every 5280-bit transmission.
- 64G FEC is always enabled as part of the Gen 7 standard, correcting up to 15 symbols in every 5440-bit transmission. A symbol consists of 10 bits, so there are 544 symbols in every 5440-bit transmission.
- 128G FEC uses RS (544,514) FEC coding to correct up to 15 symbols in every 5440-bit transmission. 128G Gen 8 Fibre Channel allows the choice of either a Single FEC or Dual Interleaved FEC that can be negotiated between link partners. If Dual Interleaved FEC is negotiated, the frame data on the link is encoded and decoded using two FEC codewords of 5440 bits each, providing enhanced error correction capabilities compared to a single FEC.

Refer to the *Brocade Fabric OS Administration Guide* for FEC configuration options and limitations.

8.3.4 Buffer Credit Loss Detection and Recovery

BBC recovery allows links to recover after losing one or more buffer credits. If a credit loss is detected, recovery initiates. BBC recovery is accomplished through a link reset in which performance is maintained; frame and BBC counters are reset. BBC recovery is supported on E_Ports, EX_Ports, and F_Ports.

BBC Loss Detection and Recovery enables Brocade hardware to detect and recover lost credits on backend ports without user intervention. Enable credit loss detection and recovery on Brocade platforms. This feature is disabled by default. BBC recovery is enabled automatically across long-distance connections where the E_Port, EX_Port, or F_Port recovery mechanism is supported.

8.3.5 RASLog Messages

RASLog messages report significant system events and information (failures, errors, and critical conditions) and show the status of high-level user-initiated actions. RASLog messages are forwarded to the console, configured syslog servers, and configured Simple Network Management Protocol (SNMPv3) traps or informs. The SANnav Management Portal server can be used as a RASLog receiver.

The following messages and descriptions are the severity levels:

- 1 = CRITICAL

  Critical-level messages indicate that the software has detected severe problems that will cause a partial or complete failure of a subsystem if not corrected immediately; for example, a power supply failure or a temperature rise must receive immediate attention.
- 2 = ERROR

  Error-level messages represent an error condition that does not impact overall system functionality significantly. For example, error-level messages might indicate timeouts on specific operations, failures of specific operations after retries, invalid parameters, or the inability to perform a requested operation.
- 3 = WARNING

  Warning-level messages highlight a current operating condition that should be checked, or it might lead to a failure in the future. For example, a power supply failure in a system with redundant supply issues relays a warning that the system is no longer operating in redundant mode.
- 4 = INFO

  Info-level messages report the current nonerror status of the system components, for example, detecting the online and offline status of a fabric port.

8.3.6 Audit Log Messages

Auditing is designed to support post-event audits and problem determination based on high-frequency events, such as security violations, zoning configuration changes, firmware downloads, and specific fabric events. Audit messages flagged as only AUDIT are not saved in the switch error logs. Audit messages can be streamed to the console and forwarded to syslog servers for further processing. Audit log messages are not sent to SNMP. There is no limit to the number of audit events that can occur.

For any given event, audit messages capture the following information:

- Date and time.
- Platform name.
- Username: The name of the user who triggered the action.
- User role: The access level of the user, such as root or admin.
- Event name: The name of the event that occurred.
- Event information: Information about the event.

8.4 Monitoring the Switches

Consider implementing some form of monitoring for each switch. Issues often start relatively benign and gradually degrade into more severe problems. Monitoring the logs for warning, critical, and error severity messages will go a long way in avoiding many problems. Consider the following actions:

- Plan a centralized collection of RASLogs and perhaps Audit logs through syslog. You can optionally filter these messages relatively easily through some simple scripting programs, or you can perform advanced correlation using an event management engine.
- Brocade platforms can generate SNMP traps for most error conditions. Consider implementing an alerting mechanism, such as SNMP or email notifications.

8.5 Latencies

Latency has many causes:

- Slow devices such as hosts and storage arrays
- Oversubscribed devices
- Long-distance links
- Servers that are not responding rapidly to previous I/O requests
- Degraded cables and failing SFPs due to I/O retries

Very little can be done in the fabric to accommodate end-device latencies, typically addressed through other means. Applications might require tuning to improve performance. Array latencies are dealt with through array and LUN reconfiguration, data migration, and technology refresh. Long-distance problems might require more bandwidth or adjustment of the switches' distance settings. Failing fiber infrastructure and SFPs must be identified and replaced. At best, Brocade fabrics can help identify problem sources. Broadcom has worked diligently to enhance Fabric OS RAS features, congruent with ever-changing customer requirements. Some of these features are described briefly in the following sections.

8.6 Misbehaving Devices

All fabrics are vulnerable to the effects of misbehaving devices, that is, a server or storage device that stops functioning correctly. The effects of misbehaving devices can be severe, causing other applications to fail intermittently, failover, or stop altogether. Broadcom has implemented several new features designed to detect misbehaving devices and isolate them from other devices in the fabric.

Isolating a single server has much less impact on applications than disabling a storage array port. Typically, a storage port services many applications, and the loss of that port can severely impact all connected applications. One of the advantages of a core-edge design is that it is straightforward to isolate servers from their storage and ensure that any action applied to a host port for a given behavior can be very different than the action applied to a storage port for the same behavior.

Detailed guidance on monitoring for misbehaving devices and configuring fabrics to respond to developing issues can be found in the *SAN Fabric Resiliency and Administration Best Practices User Guide*.

8.7 Design Guidelines

Consider the following design guidelines:

- **Transaction-based systems**: Ensure that the ISLs or UltraScale ICLs traversed by transaction-based systems accessing storage do not contain excessive flows. The fan-in from initiators should not exceed a ratio of 10 to 1. Also, ensure that there is as little interference from other applications as possible so that latencies and congestion from other sources (called perpetrator flows) do not affect the overall performance of the applications (called victim flows).
- **I/O-intensive applications**: Bandwidth is the most common constraint for I/O-intensive applications. Modern fabrics provide more bandwidth than needed, except for the most powerful hosts. Ensure that high-performing systems do not interfere with other applications, particularly if utilization spikes are scheduled at specific times or batches. Add more paths, ISLs, or trunks when in doubt.
- **Clusters**: Clusters often have behavioral side effects, particularly during storage provisioning. A cluster inundating a fabric and storage array with LUN status queries and other requests can cause fabric congestion and stress array controllers. Spread LUNs across arrays.
- **Congestion**: In some cases, traffic congestion can be remedied by adding more ISLs or BTs, assuming the congestion is between switches. In many cases, congestion occurs at a node or storage port. Brocade Gen 7 FPIN technology mitigates end-device congestion issues. Configure end devices with modern HBAs and drivers to optimize traffic across lossless Fibre Channel fabrics that use FPIN.
- **Misbehaving devices**: Little can be done in the fabric to mitigate the effects of a badly behaving device other than to remove it from the fabric. Brocade Fabric OS, Port Fencing technology, is designed to isolate rogue devices. Port Fencing works with MAPS to disable a port when a specific threshold is reached. With FPI monitoring, Port Fencing can detect and isolate high-latency devices. High-latency devices frequently impact many other devices in the fabric.
- **Initiator and targets**: Isolate host (initiator) and storage (target) ports onto separate switches for more effective management and control over misbehaving and high-latency devices. The effect on an environment is often less severe if a node port is disabled than a storage port, which services many servers.

Chapter 9: Fibre Channel Routing

9.1 Overview and Purpose

A large SAN might have thousands of end devices, which could inundate or exceed fabric scalability, fabric services, convergence timeliness, and user manageability. FCR constrains fabric services to an edge or backbone fabric. With FCR, fabric services do not extend beyond an edge or backbone fabric. Fabric services are self-contained within each edge fabric or backbone. An example of a fabric service is the name server.

Limiting fabric services to within each edge fabric is done for various reasons:

- The overall SAN can scale to a much larger relative size than the maximum scalability of the fabric services within each edge fabric.
- Within an edge fabric, FCR reduces the number of switch domains and managed zones.
- Edge-fabric disturbances and reconfigurations only affect local fabric services, thereby providing fault isolation.
- FCR increases security because end devices cannot communicate outside an edge fabric unless explicitly zoned.

9.2 Edge Fabrics

Edge fabrics are traditional fabrics, except they are connected to backbone EX_Ports. Edge fabrics contain end devices and might be connected to other edge fabrics through the backbone fabric. Backbones EX_Ports connect to edge fabric E_Ports. There are no EX_Ports in an edge fabric.

Generally, edge fabrics follow the core-edge architecture, the same as traditional fabrics. Unique to FCR is edge fabric interconnectivity through a backbone.

9.3 Inter-Fabric Links

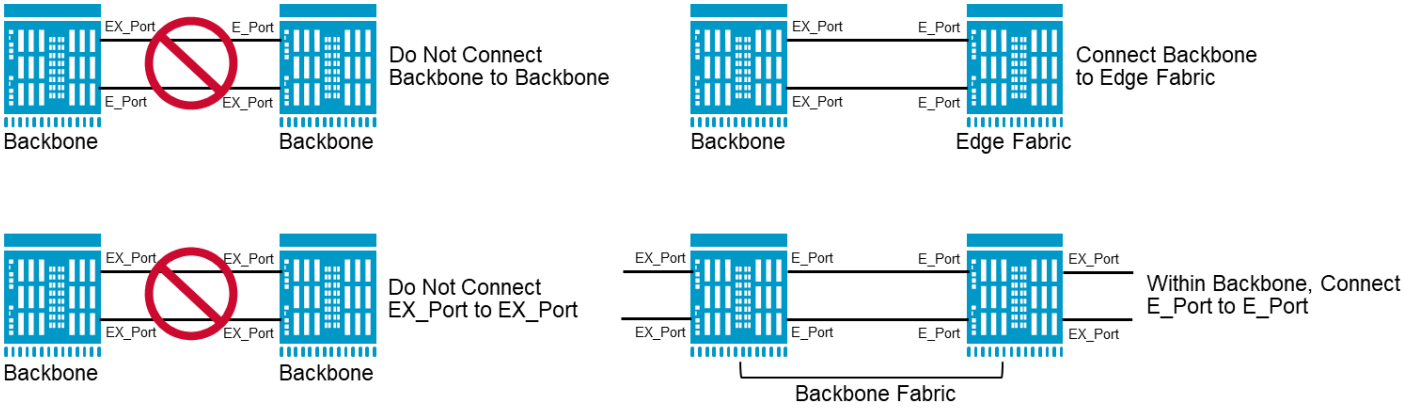
An IFL connects an EX_Port to an E_Port. It is an ISL that spans from an edge fabric to a backbone.

Provision enough IFLs between each edge fabric and the backbone to accommodate the projected peak traffic load, plus planning for IFL outages due to a bad port, optic, or cable.

9.4 Backbone Fabrics

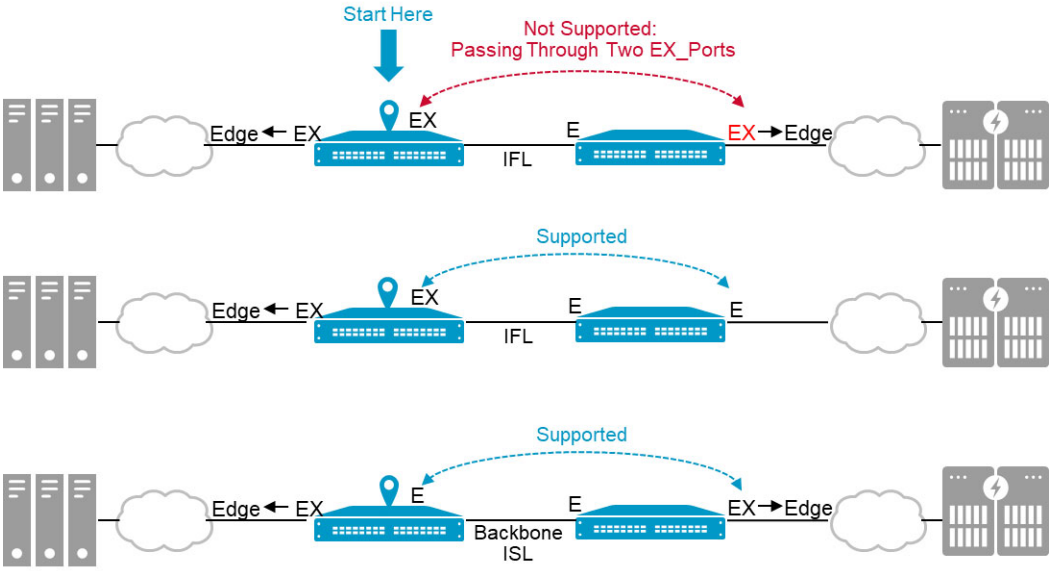
Backbone fabrics contain EX_Ports and are the boundary for fabric services. Each backbone has a unique backbone ID (BBID). A fabric can contain one or more backbones with end devices or no end devices connected. Both architectures are supported. The backbone contains the EX_Ports; EX_Ports do not exist in edge fabrics. Do not connect backbones together that have different BBIDs; see the following figure. A backbone fabric can contain extension links, which are considered an ISLs; see FCR and Extension.

Figure 10: Supported Backbone Architectures



EX_Ports are fabric services boundary points, and fabric services do not pass beyond an EX_Port. Two EX_Ports cannot be connected. EX_Ports only connect to E_Ports, and the EX_Ports must exist on the backbone side, not the edge fabric side. Topology supportability is determined by starting within an Fibre Channel router and moving to the end device. Traffic cannot pass through more than one EX_Port along the path. The architecture is unsupported if more than one EX_Port is traversed, see the following figure.

Figure 11: Supported FCR Architectures



There are many factors to consider when designing backbone fabrics. Backbone fabrics vary based on size, requirements for redundancy, and distance between edge fabrics. Generally, SAN architecture recommendations apply equally to backbone and storage fabrics. There should be redundant fabrics, and each fabric should have redundant paths to every edge fabric. Consider the following factors when identifying the best switch platforms and backbone topology, including interconnections. The number of edge fabrics can impact the backbone topology and how they attach. Brocade FCR can be enabled on standard Fibre Channel ports, but a license might be required.

Composition of edge fabrics:

- **Scale and interoperability**: Ensure that director and switch platforms can support the scale and interoperability needed.
- **Legacy SAN platforms**: Anywhere in the SAN, earlier directors/switches or firmware might impact supported features, manageability, and interoperability.
- **Advanced SAN applications and features**: Some advanced SAN applications and features might not be compatible with FCR or a particular platform type.

Projected inter-fabric traffic patterns:

- **Quantity (bandwidth utilization)**: Provision enough ISLs within the backbone to accommodate projected peak traffic loads that traverse the backbone.
- **Bursty versus peak traffic**: Bursty traffic is a sudden spike that rapidly dissipates. It is not the same as peak traffic, which might not be bursty. Infrequent, bursty traffic can be forgiving. Traffic bursts might cause temporary response time increases due to congestion. Buffer credits might be withheld until a burst subsides. Such congestion is less likely with traffic patterns of a continuous nature.
- **Small versus large frame size**: Fibre Channel is a high-speed, low-latency protocol. It relies on BBC flow control. This mechanism is a fundamental part of Fibre Channel and provides lossless data communications. A sequence of small frames uses the same number of BBCs as a series of large frames. On the other hand, large frames use more bandwidth. In other words, a large amount of small-frame traffic can fully utilize available buffers while consuming only a minimal amount of bandwidth. Therefore, consider not only bandwidth but also the typical frame size. For instance, Fibre Channel compression creates primarily smaller Fibre Channel frames. If the bulk of frames is expected to be smaller, additional buffers should be allocated to the paths handling those I/O patterns. Pay extra attention to this type of congestion because congested backbones adversely impact the performance of all connected edge fabrics. When in doubt, overprovision IFLs.
- **Distance (location of fabrics)**: Long-distance IFLs require adequate bandwidth and BBCs to prevent data transmission congestion and droop, respectively. Consider all potential traffic flows that might traverse the long-distance links. Long-distance links have more latency, simple physics time = distance/rate. Therefore, overprovisioned long-distance links might prevent oversubscription such that unexpected bursts do not adversely impact flows.
- **Virtual Fabrics**: All EX_Ports must reside in the base switch. The base switch does not support ISL R_RDY mode. If a logical switch has XISL enabled, you cannot connect an EX_Port to that logical switch. The base switch is similar to a backbone switch, and a base fabric is like a backbone fabric. All switches in a backbone fabric must have the same backbone fabric ID, which must be unique relative to any edge fabric.

Potential growth:

- **Number of fabrics**: If the number of fabrics is likely to increase, then deploy backbone fabrics to readily accommodate additional edge fabrics and additional traffic loads.
- **Size of fabrics**: If the size of edge fabrics is likely to grow, and the inter-fabric traffic is expected to grow accordingly, provision additional IFLs and ISLs such that the capacity of available paths stays well ahead of current usage. That way, incremental growth on the edge can be accommodated without immediately upgrading the backbone.
- **Amount of traffic between fabrics**: If the inter-fabric traffic is expected to grow even without growth in the individual edge fabrics, then provision additional IFLs and ISLs such that the capacity of available paths stays ahead of current usage. That way, incremental increases in data flow across the backbone can be accommodated without immediately upgrading the backbone. Make sure that you allow for plenty of room for backbone expansion.

NOTE: Refer to the *Brocade SAN Scalability Guidelines* for FCR scalability limits.

Consider using FCR under the following conditions:

- There are requirements for added scalability.
- There are benefits to compartmentalizing manageability.
- Enhanced security is required.
- There is a limited number of initiator-target pairs shared between edge fabrics.
- There is a limited number of LUNs shared between edge fabrics.
- Archiving devices, such as tape libraries, must be shared.

The implementation and configuration of IFLs in the case of FCR, should be based on the expected data volume between the backbone and edge fabrics and the desired level of redundancy. Some architectural examples of FCR topologies follow.

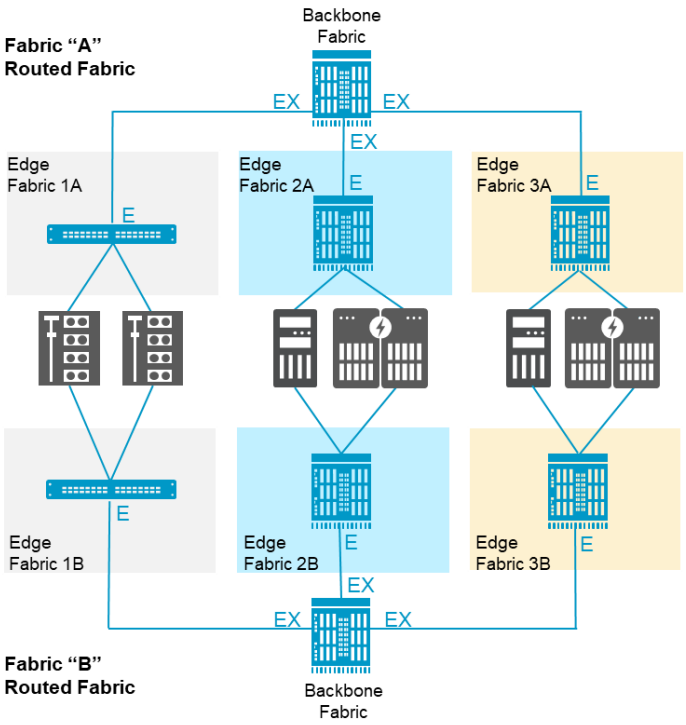
Except in the case of tape, which often has only a single pathway, there should always be A and B fabrics, each with IFL redundancy. A routed fabric environment consists of one or more edge fabrics interconnected by one or more backbone fabrics. Multiple backbone fabrics are parallel and belong to only the A or B fabric, not both. A backbone fabric can be a single switch, multiple switches, or a core-edge topology. These topologies are valid for the edge fabrics as well.

In Figure 12, the architecture consists of three edge fabrics and a backbone fabric. A and B fabrics are shown. The *A* backbone connects to each edge fabric through EX_Ports. EX_Ports in the backbone connect to E_Ports in the edge fabric to form IFLs. Each backbone must have a unique backbone fabric ID (BBFID), and all switches within that backbone must have that same BBFID. The default is 128, and when a single backbone is deployed, as shown in the following figure, no BBFID needs to be configured because the default will suffice. An alias can be assigned to BBFIDs.

Each edge fabric must have a unique edge-fabric ID (EFID), and all EX_Port connections to that edge fabric must use that EFID. Each EX_Port is configured with the corresponding EFID belonging to the edge fabric that it connects. E_Ports are not configured with any additional parameters when connecting to EX_Ports.

A collapsed-core backbone is a relatively straightforward FCR architecture.

Figure 12: Routed A and B Fabric Collapsed-Core Architecture



In the following figure, a separate backbone fabric is not deployed. Instead, the middle fabric is assigned as the backbone, and end devices connect directly to the backbone. There are three fabrics, each with its own self-contained fabric services.

Not having a separate backbone fabric limits the topology from being an interconnected full-mesh. There are only two connections coming out from the center edge fabric, and there is no connection between the left and right edge fabrics. Such a design violates the previously mentioned supported FCR architectures by creating a situation in which more than one EX_Port might be traversed from inside an Fibre Channel router to the ultimate destination device.

This design might be used when the cost of an additional fabric for the backbone is prohibitive.

Figure 13: Common-Backbone, Dual Collapsed-Core Architecture

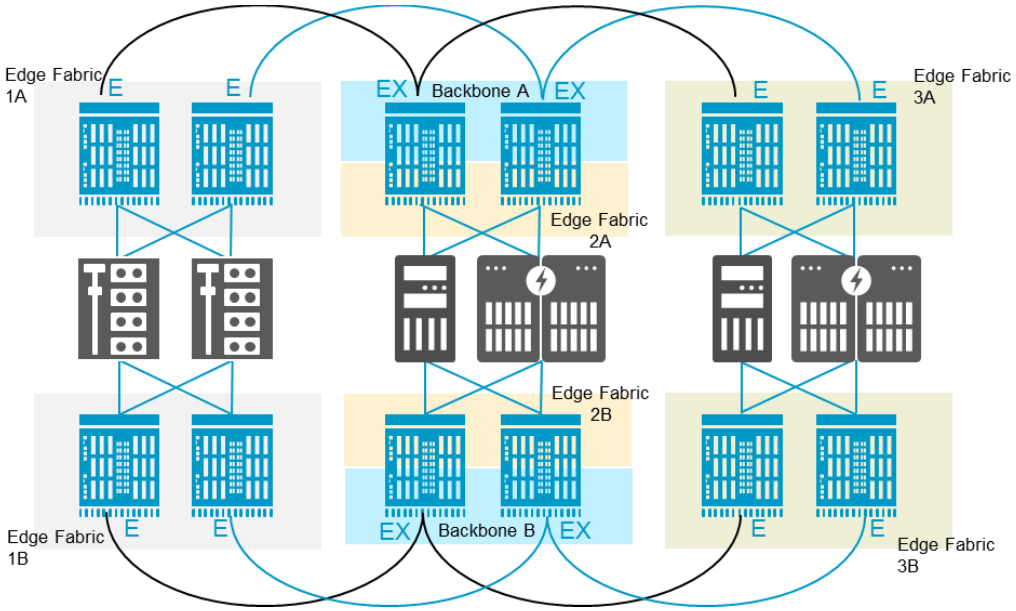
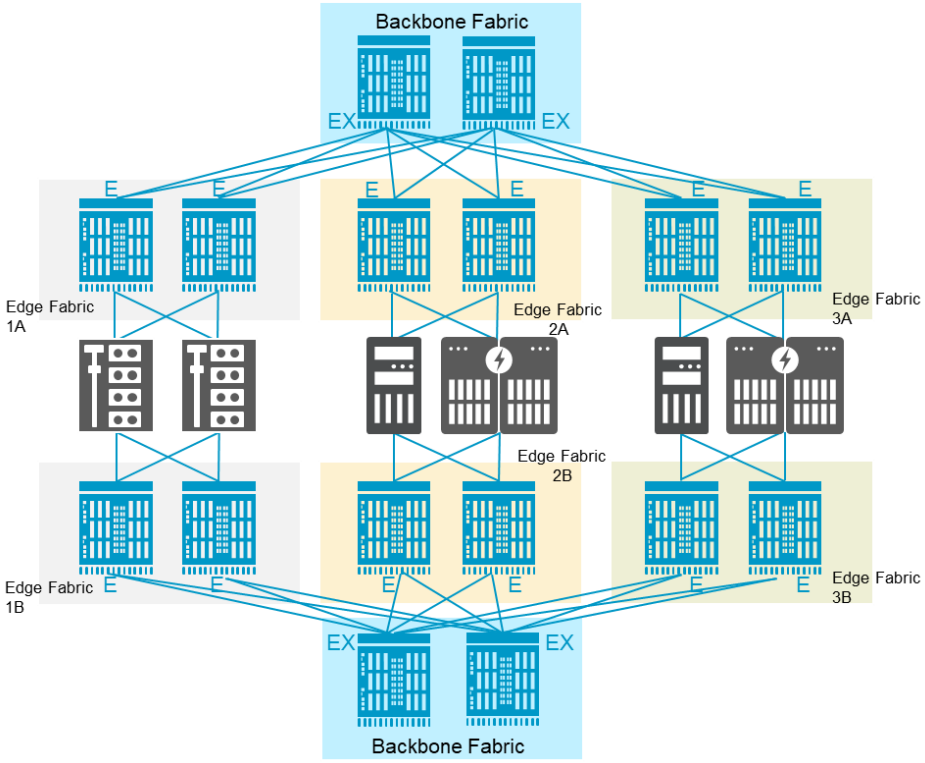


Figure 14 shows a routed SAN with A and B fabrics, each having a dual-core backbone and a unique BBFID. The EX_Ports are exclusively in the backbone, and fabric services do not pass beyond the EX_Ports. There are three edge fabrics, each with its own EFID and multiple IFLs. The dual-core backbone architecture is highly redundant, resilient, and scalable for critical enterprise applications demanding zero downtime. Considering the dual-core backbone's scalability, it is relatively easy to manage operationally.

Figure 14: Dual-Core Backbone Routed Fabric



9.5 Redundancy

Consider the following steps to achieve FCR SAN redundancy:

- Using best practices within the edge fabrics (core-edge or collapsed-core architectures).
- Using best practices within the backbone fabrics (core-edge or collapsed-core architectures).
- Deploying dual backbone fabrics for each fabric (A and B). The need for redundancy versus cost and operations must be considered. Ask yourself what the purpose of the routed SAN is. What happens if the routing between edge fabrics goes offline, yet the edge fabrics remain online?
- Parallel IFLs between the backbone and edge fabrics, including ports, optics, and cable redundancy.

9.6 Avoiding Congestion

Bandwidth and potential utilization between endpoints must be evaluated similar to any traditional fabric, by calculating traffic flows in and out of every edge fabric and providing enough backbone bandwidth. For improved utilization and resiliency, the same best practice ISL guidelines are used to connect edge fabrics with IFLs. Higher-performance edge fabrics versus an underperforming backbone can result in an oversubscribed backbone, leading to congestion, higher latency, and longer storage response times during peak loads. If the edge fabric has 64G Fibre Channel ISLs, the backbone fabric must also have 64G Fibre Channel ISLs. Before upgrading an edge fabric, upgrade the backbone to avoid congestion and oversubscription issues.

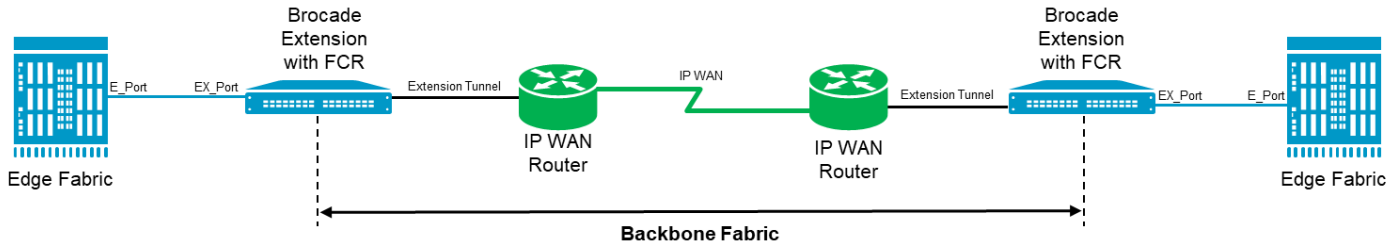
9.7 Available Paths

An optimal approach is to have multiple BT paths between edge fabrics to spread traffic across available resources. Never attach both A and B fabrics to the same backbone device. Connecting A and B edge fabrics to the same backbone device destroys the air gap between A and B and is not considered a redundant architecture and best practice. From the perspective of Fibre Channel, you should adhere to the concept of an air gap from host to storage. A standard device connected to A and B fabrics can cause a SAN-wide outage. If an air gap is implemented, faults on one fabric cannot affect the other fabric. These faults can manifest from defects in hosts, drivers, the fabric operating system, the fabric hardware, the storage hardware, the storage software, and human error. It is not relevant that FCR keeps fabric services separate because faults within one large routed fabric can transcend FCR, causing the entire SAN to fail.

9.8 FCR and Extension

FCR can be used within a single data center, across campus data centers, and between edge fabrics connected by FCIP over a metropolitan area network (MAN) or wide-area network (WAN); see the following figure. A Brocade Extension tunnel is an ISL (VE_Port to VE_Port). A VE_Port is an E_Port that is an endpoint of an Extension tunnel. Each Extension platform becomes part of the backbone fabric. EX_Ports on the Extension platforms connect to the edge fabrics through one or more IFLs.

Figure 15: Fibre Channel Routed Fabric over Extension



More information about Extension can be found in Chapter 12.

9.9 FCR Design Guidelines and Constraints

The following items are some of the key metrics and best practices for routed SAN topologies:

- Keep A and B fabrics separated from host HBA to storage ports from an FC perspective, this separation is referred to as an air gap. Air gaps do not include FCIP because, in an IP network, Ethernet switches and IP routers do not merge as Fibre Channel fabrics do. Extension VE_Ports should never connect fabric A to fabric B. This connection is the same as cross connecting a traditional ISL, which connects fabric A to fabric B.
- Localize traffic within an edge fabric to the greatest extent possible.
- Have a predefined schema for assigning domains within the SAN. For example, edges, cores, switches, EFIDs, translate domains, and BBFIDs should be within specific ranges to avoid domain overlap.
- Consider upgrading backbone fabrics before upgrading edge fabrics to avoid oversubscription and congestion.
- During regular operation, have no more than one long-distance ISL or extension tunnel between the source and destination. During an outage, an additional hop might be used for high availability. In a triangle architecture where the primary link is down, the remaining legs can be used as the backup path; however, latency and response times might be longer.
- Long-distance links are within the backbone, not between an edge fabric and the backbone. Edge fabrics are isolated from disruption because fabric services are not extended beyond an EX_Port. Most often, long-distance links are the primary cause of instability.

- Logical SAN (LSAN) zones are required for end devices communicating from edge fabric to edge fabric across a backbone. In other words, do not make zones within edge-fabric LSAN zones.
- For each fabric, fully redundant backbones improve resiliency. Fabric A would not be impacted if one of its backbones failed, the same with fabric B. Both fabric A backbones must fail before fabric B is entirely relied on to maintain operations.
- Redundant backbone fabrics connected to the same edge fabric must have unique BBFIDs. Refer to the case where there are redundant fabric A backbones and redundant fabric B backbones. There are no cross-connections between A and B fabrics, nor are there cross-connections between the parallel backbones within fabric A or fabric B.

Chapter 10: Virtual Fabrics

Virtual Fabrics (VF) is an architecture to virtualize hardware boundaries within a platform. Traditionally, fabric design and management are done at the granularity of a physical switch. VF allow fabric design and management to be done at the granularity of a port.

VF is a suite of related features to customize logical fabrics based on requirements. VF consist of the following specific features:

- Logical switches
- Logical fabrics
- Device sharing

Hardware-level fabric isolation is accomplished through VF, which partitions ports into one or more logical switches.

ISL-connected logical switches form logical fabrics. As port density grows, switch partitioning enables storage administrators to divide physical switches into multiple logical switches. Without VF, a Fibre Channel switch is limited to 512 ports.

There are multiple ways to connect logical switches: a traditional ISL, an extension ISL, an IFL (EX_Port used by FCR), or an extended ISL (xISL, multiple LF tagged traffic between two Base Switches. An ISL is used for connecting Fibre Channel traffic between logical switches.

An ISL carries data traffic within the logical fabric of which the ISL is a member. An advantage of VF is that multiple logical switches can share a common physical ISL, called an xISL. Each logical switch does not require a dedicated ISL (DISL). For multiple logical fabrics to share an ISL, Virtual Fabrics supports xISL connections. An xISL is a physical connection between two base switches. Base switches connected by an xISL form a base fabric. A base switch is a logical switch used for intra-fabric and inter-fabric communication.

Once a base fabric is formed, VF determines the best route between all associated logical switches and logical fabrics. For each local logical switch and every destination reachable through the base fabric, a logical ISL (LISL) is created. Thus, an xISL is a physical link between base switches, carrying multiple logical ISLs. In addition to xISLs, a base fabric supports EX_Ports for connectivity to edge fabrics using FCR. An FCR link between an EX_Port and an E_Port is called an IFL. Base switches interoperate with FCR through EX_Ports in the base fabric or EX_Ports in a different backbone fabric.

10.1 Use Case: FICON and Open Systems (Intermix)

VF enables customers to share FICON and FCP (SCSI and NVMe) traffic on the same physical platform. As chassis densities increase, this is a viable option for improved hardware utilization while maintaining director-class availability. The following items are the primary reasons for moving to an intermix environment:

- Array-to-array Remote Data Replication (RDR) of FICON volumes. Most array replication uses FCP for FICON volumes.
- ESCON-FICON migration
- Sharing of infrastructure in a nonproduction environment
- Reduced TCO
- Growth of zLinux on the mainframe

From a SAN design perspective, consider the following guidelines when considering FICON intermix:

- Connect devices across port blades, connectivity from the same device should be spread over multiple blades.
- A one-hop-count architecture applies; however, there are *hops of no concern* in some cases. Refer to the *Brocade FICON/FCP Intermix Best Practices Guide* for details.

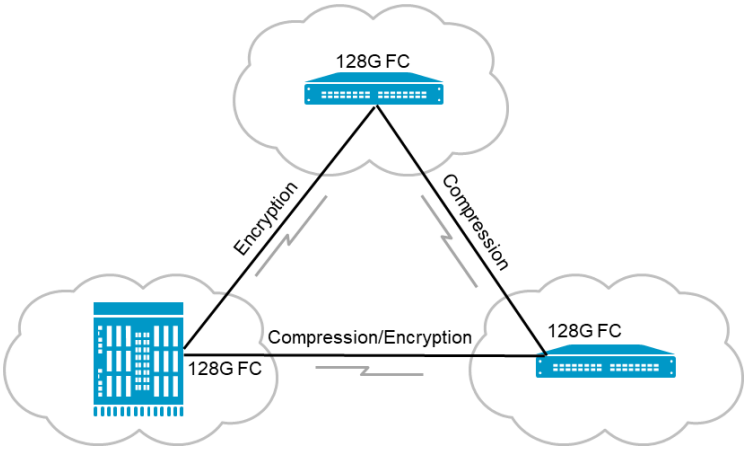
Chapter 11: Intelligent Fabric Services

11.1 In-flight Encryption and Compression

Brocade Gen 6 and later platforms support in-flight compression and encryption at the port level for local and long-distance ISL links; see the following figure. In-flight data compression is a valuable tool for saving money when bandwidth caps or usage charges limit data transfer between networks. Similarly, in-flight encryption provides an additional layer of security when transferring data between data centers. Starting with Fabric OS 10.x, additional quantum-resistant algorithms per CNSA 2.0 are supported. Also, as of Fabric OS 10.x, MD5, SHA1, and DSA are deprecated.

Refer to the *Brocade Fabric OS Administration Guide* for the latest information

Figure 16: Latency for Encryption and Compression



As the frame is processed, enabling in-flight ISL data compression or encryption increases ASIC latency. At each stage of a Gen 7 fabric running at 64G, including encryption, compression, and local switching, the approximate latency is 6.2 microseconds. Actual latency will vary depending on the switch product generation and data rate.

11.1.1 Virtual Fabric Considerations: Encryption and Compression

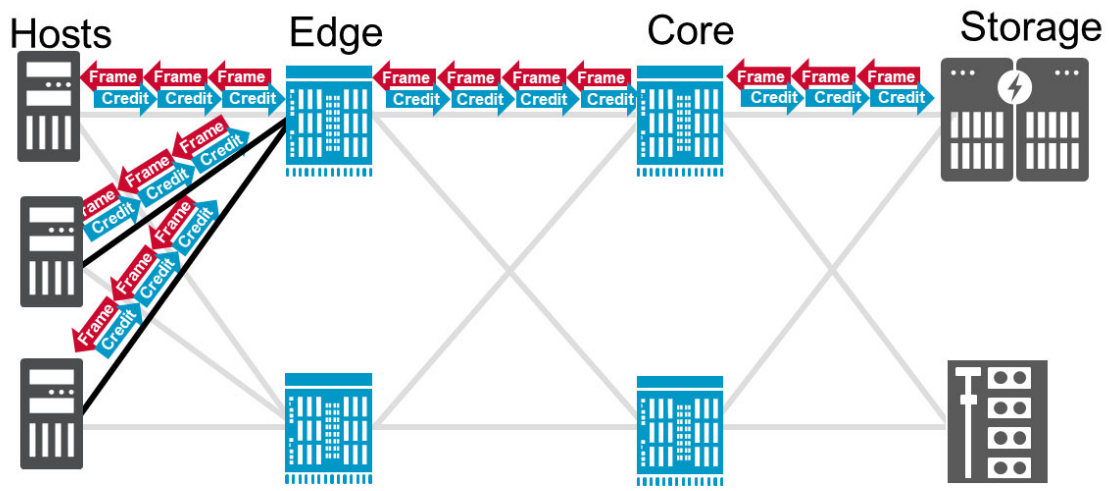
E_Ports in logical switches, base switches, or default switches can support encryption and compression. Both encryption and compression are supported on xISL ports but not on LISL ports. If encryption or compression is enabled and ports are moved from one LS to another LS, the encryption and compression must be disabled before moving to another LS.

11.2 Fabric Notifications

Fibre Channel networks can be difficult to troubleshoot because flows are difficult to visualize, and the affected devices are not likely to correspond with the problem cause. Fibre Channel uses a credit-based flow-control mechanism (see Figure 17), with inherent congestion characteristics due to head-of-line blocking. Broadcom introduces a hardware, software, and management solution called Fabric Notifications for reducing and eliminating congestion.

Collecting transport characteristic data from various sources, evaluating it, and disseminating it to interested devices allows for faster and sometimes automatic problem resolution. End devices can employ primary response and recovery mechanisms. Fabric information is helpful for end devices, which have useful information for the fabric and peer end devices. Fabric Notifications are crucial in collecting and disseminating information among related and interested devices.

Figure 17: Freely Moving Lossless Credit-Based Flow-Control Fibre Channel Network



Fabric Notifications addresses four issues: congestion (oversubscription and credit stall), link integrity, and SCSI command delivery failure.

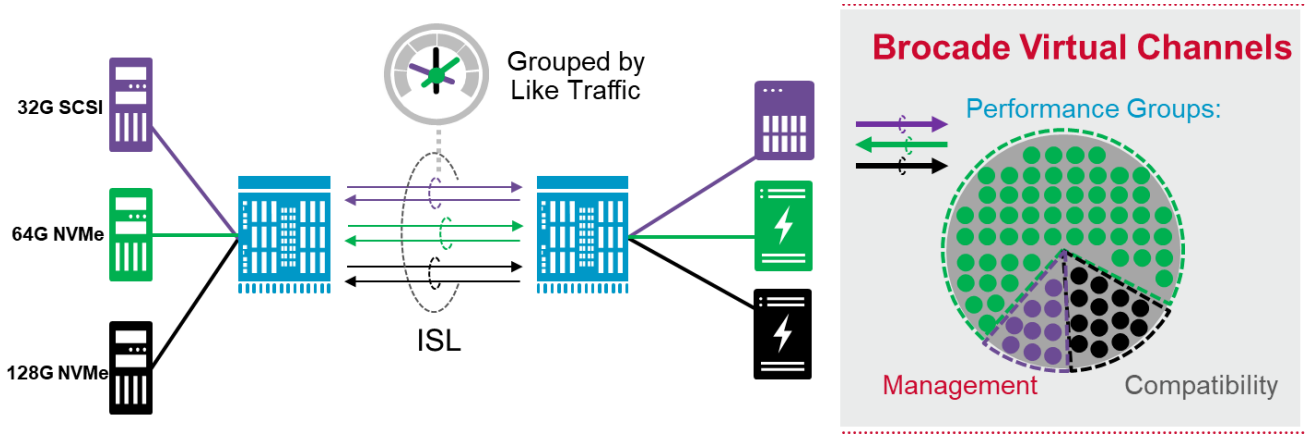
11.3 Adaptive Traffic Optimizer

Brocade VCs worked well on earlier generation platforms at 1, 2, 4, 8, and 16-gigabit rates by leveraging multiple logically independent paths. However, those independent paths were not selected based on destination port speed. To a degree, virtual channels mitigated interference between slower flows impeding faster flows. For example, an 8Gb/s flow could impede a 16Gb/s flow. Optionally, critical faster flows could be manually assigned to a high QoS VC, and slower flows could be manually assigned to a low priority QoS VC to prevent such interference.

Technology evolves, and Broadcom has optimized VC efficiency by enhancing effectiveness to targeted flow characteristics. Demands on an enterprise SAN have never been more significant due to other complimentary technology evolutions such as NVMe, all-flash array (AFA), and host virtualization. Flows compete for resources, and head-of-line blocking is not an option and must be efficaciously dealt with, which is where Traffic Optimizer helps.

Adaptive Traffic Optimizer is available with Brocade Gen 8 platforms. Adaptive Traffic Optimizer is backwards compatible with the standard Traffic Optimizer available with Brocade Gen 7 platforms, and it automatically learns and adapts to changing application demands by dynamically load balancing traffic across VCs; see the following figure. Adaptive Traffic Optimizer organizes and manages traffic flows using performance groups, and fabric resources are allocated based on performance groups. Flows are assigned to a performance group based on the destination port speed and protocol (SCSI or NVMe). Brocade fabrics know the destination port speed and protocol for every flow.

Figure 18: Host to Storage through Traffic Optimizer VCs



11.3.1 Adaptive Traffic Optimizer

On the Gen 7 Brocade platforms, Traffic Optimizer Performance Groups were defined based on a static policy. In Gen 8, Adaptive Traffic Optimizer is a Traffic Optimizer technology enhancement. Adaptive Traffic Optimizer automatically learns all flows in the fabric. It then creates performance groups to serve all speed and protocol combinations, allowing a more effective allocation of VCs and performance groups based on the actual traffic characteristics in the fabric. A validation is performed periodically based on Adaptive Traffic Optimizer learning; performance group composition across the fabric is altered if needed. Log messages indicate the start and completion of an optimal performance group set.

There are no changes to QoS zones, SDDQ, or oversubscription.

Chapter 12: Extension

The extension content has been removed from this document. Extension is a considerable subject, and a separate extension document exists. The content from this Chapter was added to and will be managed in the extension document. Please refer to the document: *Design Guide: Brocade Extension Best Practices*.

Chapter 13: SAN Design for Critical Workloads

All-flash arrays (AFA) are the technology standard in enterprise data centers. As transitioning to FC-NVMe is underway, critical business applications increasingly depend on consistent low-latency, and high-throughput storage performance for demanding performance-sensitive workloads. When designing a SAN, it is vital to consider the placement of critical workloads relative to storage, the fan-in ratio to storage ports, and Trunks.

Protecting critical workloads is crucial. Brocade technology provides measures such as Adaptive Traffic Optimizer, FPIN, MAPS, and SDDQ to avoid workload interference that might experience congestive behavior. Ideally, the most demanding and critical workloads have dedicated storage ports, maybe a dedicated array, and the shortest possible path through the fabric. The purpose is to avoid other workload interference resulting in congestion or backpressure, which could adversely impact the performance of critical applications.

13.1 Placement of Servers with Business-Critical Workloads

With core-edge SAN designs, connecting critical workload servers directly to the core alongside the storage ports is often advantageous. This practice works well when the number of business-critical workloads is easily defined and limited to a subset of servers, and an adequate number of core ports are available.

Suppose the number of business-critical servers exceeds the number of available ports on the core; in that case, connecting the business-critical servers to the edge switches is necessary. The most common model uses dedicated edge switches for business-critical servers to remove competing flows and decrease the fan-in ratio of servers to ISLs.

An alternative is to evenly distribute business-critical servers across the edge switches, assuming that workloads even out with other less demanding workloads. Although a logical approach, the practice has demonstrated that using this model is operationally complex, making it challenging to guarantee optimal performance for business-critical workloads.

13.2 Business-Critical VMs

In today’s data centers, it is common for business-critical workloads to run on VMs. Combining this with a digital society, the value, criticality, and performance requirements for a given application change throughout the application’s life. Inevitable change means predicting future requirements can be difficult or impossible. Placement planning from the beginning is not simple, but hypervisors can relocate VMs without disruption and migrate storage when necessary. The same principle applies to bare metal server placement, deploying dedicated hypervisor clusters connected to the core or high-performance edge switches.

VM Insight provides visibility into each VM’s workload, even on the same datastore. It enables storage administrators to monitor VM-level application performance and set baseline workload behavior. This information determines whether the SAN is the source of performance anomalies. Storage administrators can plan placement and provision based on application requirements and fine-tune infrastructure to meet service-level objectives.

Chapter 14: Access Gateway and NPIV

This chapter covers Access Gateway and N_Port ID Virtualization (NPIV) design considerations, primarily related to increasing fabric density and scale. In addition, there are descriptions for Access Gateway default port mapping, port mapping for specific architectures, ensuring balance and failover, and best practices.

Refer to the *Brocade Fabric OS Access Gateway User Guide* for detailed information about configuring and deploying Access Gateway.

Standards-based NPIV connects multiple virtual F_Ports to a single physical N_Port. These virtual F_Ports connect multiple host initiators, such as a hypervisor with virtual HBAs or a storage device with multiple virtual targets, to a physical HBA port. NPIV was initially developed to provide access to Fibre Channel devices from IBM mainframes and improve the efficiency of mainframe I/O for virtualized environments.

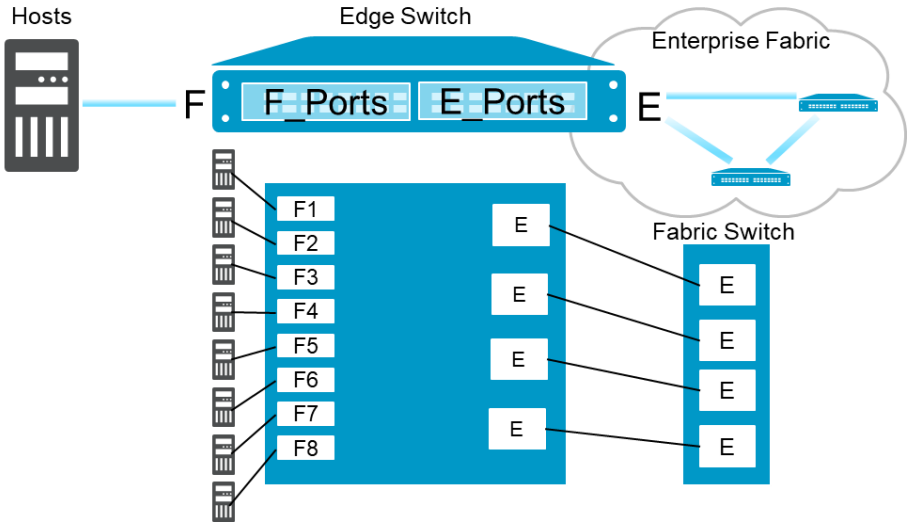
A use case is a switch configured in NPIV mode and connected as an Access Gateway to the fabric. Access Gateway does not participate in the fabric as a domain; it extends fabric ports without adding a domain. A switch in NPIV mode is a way to connect another vendor's switch. Broadcom supports connecting other standards-based switches in NPIV mode.

The following items are common NPIV use cases:

- Using Access Gateway to increase port count without increasing domains
- Deploying Access Gateway in a POD architecture
- Connecting many embedded switches that are in blade-server chassis
- Connecting other vendor switches (Cisco UCS FI)
- Creating storage arrays with a virtual storage controller architecture that presents separate virtual target ports behind the same physical target port
- Provisioning hypervisors with virtual HBAs to VMs for Raw Device Mapping (RDM) storage allocation

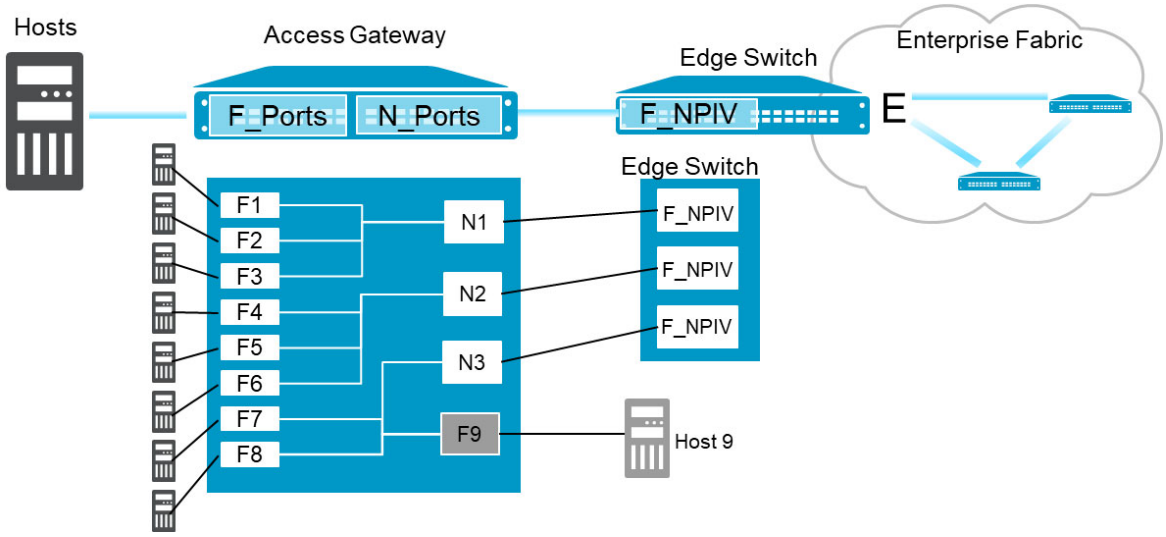
As shown in the following figure, a switch in Access Gateway mode connects F_Ports to the fabric as N_Ports instead of E_Ports. The following figure shows a switch in native mode with all devices connecting to F_Ports and switch-to-switch connections as E_Ports (ISLs).

Figure 18: Switch Functioning in Native Mode



The following figure shows a switch in Access Gateway mode with all devices connecting to F_Ports, then mapping to N_Ports. The Access Gateway N_Ports connect to the fabric's NPIV F_Ports, typically on an edge switch.

Figure 19: Switch Functioning in Access Gateway Mode



Switches in Access Gateway mode are transparent to the host and fabric. The hosts accessing the fabric can be increased without increasing domains. Access Gateway mode simplifies configuration and management by reducing the number of domains. Fabric-specific configuration is inherited on fabric switches; for example, zoning. Access Gateway does not participate in these fabric services.

Placing switches in Access Gateway mode does not consume a fabric domain ID. The main reason for using Access Gateway mode is to achieve scalability with many small switches. The embedded switches can quickly approach the maximum domain limit in a blade server environment.

Access Gateway functionality is enriched, although there are scenarios in which full switch functionality is advantageous. Deciding to use Access Gateway involves evaluating whether Access Gateway is an appropriate option. Identifying and isolating misbehaving devices in a fabric with many legacy devices is easier in a complete switch environment.

For configurations with hosts and targets connected to the same Access Gateway, traffic must first pass through the Access Gateway to a fabric switch. If Access Gateway is not used, local traffic is switched by the embedded switch and does not need to traverse the Access Gateway to the fabric and back. The theoretical domain limit in a fabric is 239, but most fabrics are limited to a much smaller number. The maximum number of domains supported in Brocade fabrics is 56. The domain count limit becomes relevant when many small-port-count switches are deployed. Large blade-server deployments with embedded switches push the domain count beyond supported limits. Fibre Channel switches in blade server enclosures typically represent fewer than 32 ports.

14.1 Benefits of Brocade Access Gateway

- **Scalability:** Access Gateways can be added to a fabric without increasing domain count. A scalability constraint is avoided when small-port-count switches and embedded switches are part of the infrastructure. Registered state change notifications (RSCN) are reduced; only downstream initiators on the Access Gateway are passed to the fabric. Access Gateway upstream ports can connect to one or more fabrics. Access Gateway cascading is supported and reduces the fabric connections required to support attached hosts.

- **Error isolation and management:** Most initiator errors do not propagate through the fabric. Management activities on Access Gateway are isolated from the fabric. Disconnecting an upstream port does not cause a fabric event. Blade-server administrators manage Access Gateway. Storage administrators using NPIV, provision LUNs, and support zoning.
- **Increased resiliency:** Access Gateway supports F_Port trunking, which increases upstream resiliency to the fabric. Losing an optic, link, or cable reduces the bandwidth of the trunk. A few frames might be lost; however, no host connections are affected.
- **Other:** Hosts or HBAs can be configured to failover to another upstream link automatically should the link being used fail. Access Gateway implements advanced features such as adaptive networking, BT, HCL (hot code load), MAPS, ClearLink Diagnostics (D_Port), credit recovery, and FEC.

14.2 Constraints

The advantages of the Brocade Access Gateway are compelling, but there are constraints:

- Although the benefits are much more evident for servers, Access Gateway supports storage devices, but the traffic must flow through to the fabric, which has limitations.
- The maximum NPIV connections per upstream port is 254.
- The maximum Access Gateway per switch is limited by the number of connections the fabric switch supports.

Primary factors to consider:

- The number of devices that attach to the fabric through Access Gateway
- The number of devices per Access Gateway N_Port
- The number of devices attached to the switch and fabric. Refer to the *Brocade SAN Scalability Guidelines* for details.

The number of fabrics an Access Gateway can be connected to is limited by the number of N_Ports on the Access Gateway. Most deployments require two Access Gateway connections to each fabric. Be aware that connecting different upstream ports to different fabrics does not reduce the requirement for redundancy. All attached servers should have dual paths to storage through different fabrics and separate Access Gateway.

14.3 Design Guidelines

Use Access Gateway to deploy blade servers, many low port-count switches, or connect blade enclosures to multiple fabrics. Access Gateway separates blade enclosure management, so server administrators manage the enclosure, and storage administrators manage the fabric. Flow separation is provided through NPIV, allowing Access Gateway to be managed separately. Integrated blade-server management tools pose no risk to fabric operations.

14.4 Monitoring

Brocade Access Gateway has been enhanced to include features found in the standard version of Brocade Fabric OS, such as Port Fencing, device security policies, FPI monitoring, and SDDQ. However, monitoring and troubleshooting NPIV flows are less feature-rich than traditional flows.

14.5 Maintenance

Maintaining Access Gateway firmware levels synchronized with fabric firmware levels usually is not required. Broadcom supports other vendors' NPIV-enabled devices where firmware synchronization is not possible. Maintaining firmware levels can be significant in large fabrics with many Access Gateways. The version of Brocade Fabric OS running on fabric switches can be upgraded during a planned maintenance window, while Access Gateways are upgraded later, dramatically reducing the amount of change required during a single maintenance window.

See the *Brocade Fabric OS Release Notes* to determine if a synchronized Brocade Fabric OS upgrade of Brocade Access Gateway devices is required.

14.6 Access Gateway Mapping

When a switch operates in Access Gateway mode, you must specify the Access Gateway device's routes to direct traffic from the devices on its F_Ports to the fabric ports connected to its N_Ports. The routes must be preprovisioned. The process of provisioning routes in Access Gateway mode is called mapping. By comparison, a switch operating in Native mode determines the best routing path between its F_Ports.

You can create two types of maps: port maps and device maps. Port maps are required. Device maps are optional and assign device WWNs to N_Ports and N_Port groups. Port mapping and device mapping operate as follows.

14.6.1 Port Mapping

Port mapping ensures all traffic from a specific F_Port goes through the same N_Port. An F_Port is mapped to an N_Port or N_Port group. To map an F_Port to an N_Port group, map the F_Port to an N_Port belonging to that group.

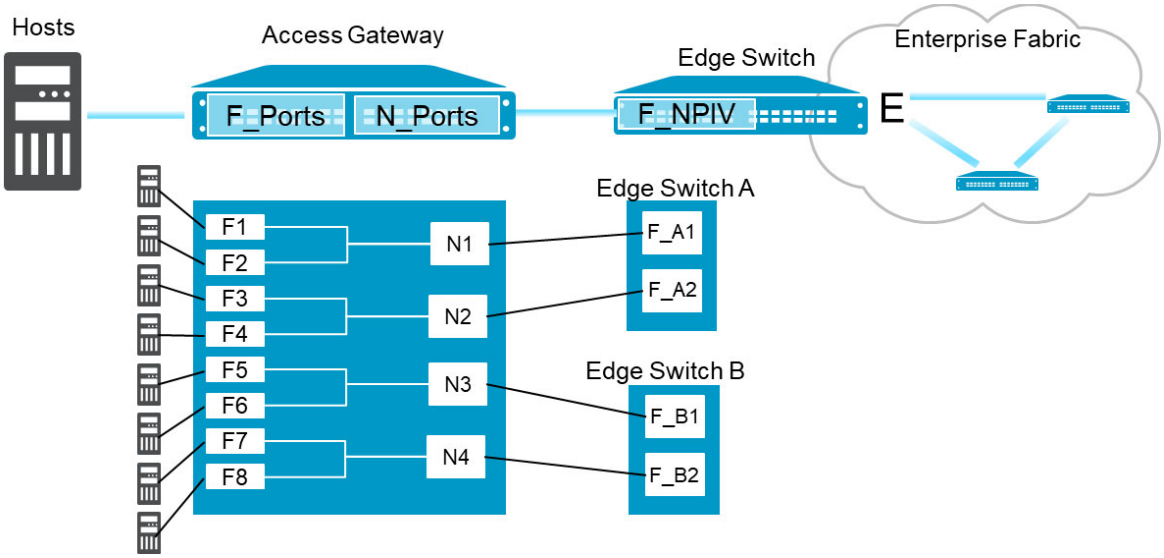
14.6.2 Device Mapping

Device mapping is optional. Port maps must exist before you can create device maps. Device mapping allows a virtual port to access its destination device regardless of the F_Port where the device resides. Device mapping also allows multiple virtual ports on a single physical machine to access multiple destinations in different fabrics. You can map a device to multiple groups. Alternatively, you can map a device to a specific N_Port.

The preferred method is to map a device WWN to an N_Port group. When a WWN is mapped to a port group, it can log in to the fabric if at least one N_Port remains online. However, when a WWN is mapped to an N_Port and a failover N_Port is specified, the WWN can only reach the fabric through the primary and failover N_Ports.

F_Ports must be mapped to N_Ports before the F_Ports can come online. The following figure shows an example where eight F_Ports are mapped evenly to four N_Ports on a switch in Access Gateway mode. The N_Ports connect to the same fabric through different edge switches.

Figure 20: Port Mapping Example



The following table shows the port mapping illustrated in the figure. F_Ports F1 and F2 map to N_Port N1; F_Ports F3 and F4 map to N_Port N2, and so on.

Table 1: Description of Port Mapping

Access Gateway		**Fabric**	
F_Port	N_Port	Edge Switch	F_Port
F1, F2	N1	Switch A	F_A1
F3, F4	N2	Switch A	F_A2
F5, F6	N3	Switch B	F_B1
F7, F8	N4	Switch B	F_B2

14.6.3 Default Port Mapping

When you enable Access Gateway mode on a switch, a default mapping is used for the F_Ports and N_Ports.

The following table describes the default port mapping for a G720. Refer to the *Brocade Fabric OS Access Gateway User Guide* for default mappings on all supported hardware platforms.

Table 2: Access Gateway Default Port Mapping for a G720

Brocade Platform	**Total Ports**	**F_Ports**	**N_Ports**	**Default Port Mapping**
G720	64	0-39, 48-63	40-47	0-6 mapped to 40 7-13 mapped to 41 14-20 mapped to 42 21-27 mapped to 43 28-34 mapped to 44 35-39, 48-49 mapped to 45 50-56 mapped to 46 57-63 mapped to 47

NOTE: By default, failover and failback policies are enabled on all N_Ports.

The default mapping can be changed to meet specific requirements for your environment. For more information, refer to the *Brocade Fabric OS Access Gateway User Guide*.

Chapter 15: Security

Many SAN security components are related to design, and deciding to use them depends on requirements rather than network functionality or performance. When designing SAN security, it is optional to implement security features. Some security features add performance overhead, others affect administrator productivity, and others have associated implementation costs. There is a balance between features, the value of protecting assets, and the chance of exploiting a vulnerability.

One clear exception is the zoning feature used to control device communication. Proper zoning is vital to fabric functionality, performance, and stability, especially in more extensive networks. Other security-related features are mechanisms for limiting access and preventing attacks on the network, often mandated by regulatory requirements and not required for operation.

This chapter describes the best practices for secure SAN communication and secure SAN infrastructure access and protection.

15.1 Zoning: Controlling Device Communication

Brocade zoning is crucial in managing device communication, essential for effective, efficient, and secure storage network use. The SAN is responsible for data flow between devices, and zoning specifies which device can communicate with another. Zoning is enforced, disallowing communication between devices not within the same zone.

Zoning protects devices from disruption by constraining the RSCN scope. Fabric changes generate notifications (RSCNs) to the fabric and end devices. Delivery is limited to devices in the zone and only when a change occurs. This limit reduces switch processing overhead by reducing RSCNs and, in rare cases, limits the impact of a faulty HBA that creates errors.

Thus, only devices in the zones impacted by a change are disrupted. Based on this information, the best practice is to create single-initiator single-target zones with one initiator and one target so that changes to initiators or targets do not impact other initiators and targets. Disruptions are minimized, as shown in Figure 21. In addition, the default zone setting should be *No Access*, devices are isolated when zoning is disabled.

Zones can be defined by either the device's connected switch port or the device's Port World Wide Name (PWWN). Although it takes more effort to use WWNs in zoning, it provides excellent flexibility. For example, moving a device anywhere in the fabric maintains a valid zone.

15.1.1 Peer Zoning

As the number of hosts increases, configuring and maintaining single-initiator zones becomes challenging. Additionally, a unique zone for each initiator and target could grow to exceed the maximum zone database size.

Many-to-one zoning defines many initiators with one target. Before the availability of peer zoning, it was common to zone multiple initiators with a single target to achieve provisioning efficiency; by today's standards, this is not a best practice.

Peer zoning is easier to manage and avoids exceeding the maximum database size; conversely, peer zoning results in increased RSCN traffic. A SAN using peer zoning provides operational efficiency and effective single-initiator zoning while reducing the database size.

Peer zoning allows a principal device to communicate with non-principal devices within the zone as a single-initiator, single-target zone. Non-principal devices can communicate only with the principal device; they cannot communicate with each other. Principal devices cannot communicate with other principal devices.

A peer zone can have one or multiple principals. In general, storage ports are assigned as principals. Multiple principal members in a peer zone are used when all the non-principals (initiators) in the zone are to share the same target (storage) ports.

The peer zone members are WWNs or aliases specifying one or more WWNs or a domain and port. You cannot mix WWNs and domain, port or associated aliases when defining peer zoning.

15.1.2 Target-Driven Zoning

Target-driven zoning is a variant of peer zoning; the user specifies the configuration in a regular peer zone, and the principal device defines the target-driven zone. The principal device is usually a storage array, but it does not have to be. Target-driven zoning must be enabled on the F_Port connected to the principal device. Target-driven zoning uses a third-party interface to manage the device and switch interactions.

Refer to the vendor's principal device manual to determine the commands and options to construct a target-driven peer zone.

Refer to the *Brocade Fabric OS Administration Guide* for additional details and considerations.

15.1.3 Zone Management: Duplicate WWNs

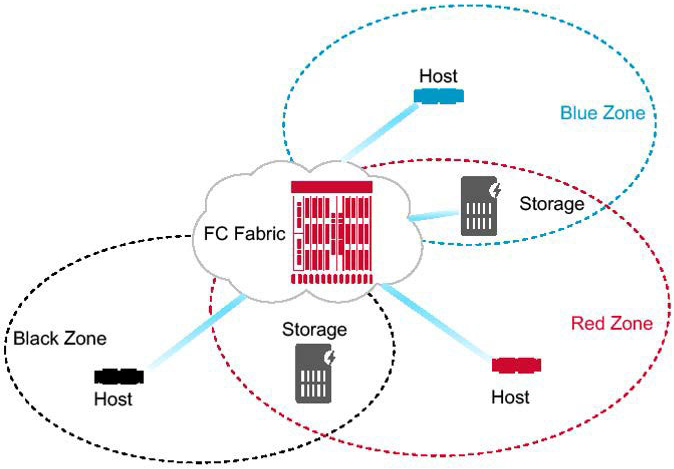
In a virtual environment like VMware or HP Virtual Connect, it is possible to encounter duplicate WWNs in the fabric, often as a transient condition. Duplicate WWNs impact switch responses to fabric service requests, such as *get port WWN*, which results in unpredictable behavior. Additionally, it represents a security risk by enabling the spoofing of the intended target. The fabric's handling of duplicate WWNs is not meant to be an intrusion detection tool but rather a recovery mechanism. When a duplicate entry is detected, a warning message is sent to the RASlog, and no effort is made to prevent the device login of the second entry.

The handling of duplicate WWNs is as follows:

- Same switch: The choice of which device stays in the fabric is configurable. The default is to retain the current device.
- Same WWN on two switches: remove both entries.
- Zoning recommendations include the following:
  - Always enable zoning
  - Use peer zoning or single-initiator zoning
  - Define zones using device PWWN
  - Set default zoning to **No Access**

Follow the vendor guidelines for preventing the generation of duplicate WWNs in a virtual environment.

Figure 21: Example of Single-Initiator to Single-Target Zones



15.2 Securing the SAN Infrastructure

An operational advantage of a Brocade SAN is quickly adding new switches to a fabric. A SAN administrator need only connect a new switch to an available port on an existing switch through an ISL and then power up the new switch. A unique domain ID is automatically assigned, and the configuration files are downloaded to the new switch. However, from a security perspective, this time-saving administrative ease-of-use capability also means that anyone with a switch and physical access could connect to an existing fabric and gain control. If an attacker with admin access on a rogue switch were to use this technique, the attacker would now have admin privileges for the entire fabric.

There are several layers of defense available to secure the SAN. In order of ease of deployment, the following list describes best practice configurations to secure the SAN. Deploy the first four layers. Deploy the remaining layers depending on the organization's security requirements.

The security defense layers in the order of deployment ease:

- Persistently disable unused ports
- Prevent switch ports from becoming E_Ports
- Configure auditing
- Use a strict fabric-wide consistency policy where possible
- Use the SCC policy to restrict switch connections to the fabric
- Use a FCS policy to restrict security configuration changes further
- Use DCC policies to restrict device access by WWN and physical switch ports
- For more sensitive environments, use DH-CHAP to authenticate devices that join a fabric

The first and simplest line of defense is to persistently disable all unused ports, which can prevent someone without administrative privileges from connecting to the fabric. It is vital to use the persistent disable option to ensure that disabled ports remain disabled after a reboot or power cycle. Otherwise, an attacker could unplug the switch to re-enable unused ports.

The second line of defense is to prevent ports from becoming E_Ports. If an unused port remains enabled, a new switch would not be able to join the fabric if the port cannot become an E_Port.

The third line of defense is configuring auditing to ensure visibility into any unexpected access, changes, events, and attacks. The fourth line of defense is to use a fabric-wide consistency policy to ensure that all fabric switches avoid a weak link exploitable attack. Subsequent layers use an access control list defense. See the following section for more information.

Implementing all lines of defense is often unnecessary. The layers implemented depend on an organization's requirements, data sensitivity, environment, and risk tolerance. In practice, only some organizations implement all levels. It is up to each organization to establish the acceptable risk and decide which features should become part of operations.

15.3 Access Control Lists (ACLs)

ACLs provide network security by enforcing a policy set. Brocade Fabric OS provides several ACL policies, including a SCC policy, a FCS policy, a DCC policy, an IP Filter policy, and others. The following subsections briefly describe each policy and outline the basic guidelines.

A more in-depth discussion of ACLs can be found in the *Brocade Fabric OS Administration Guide*.

15.3.1 Switch Connection Control (SCC) Policy

The SCC policy restricts fabric elements from joining a fabric, particularly Brocade Fabric OS platforms. Only switches specified in the policy are allowed to join the fabric. All other attempts to join will fail authentication, resulting in the E_Ports being segmented due to a security violation.

Use the SCC policy in environments where strict control of fabric members is required. Since the SCC policy can prevent switches from participating in the fabric, reviewing and adequately maintaining the SCC ACL regularly is essential.

15.3.2 Fabric Configuration Server (FCS) Policy

Use the FCS policy in environments that require strict control over fabric settings. FCS restricts fabric-wide settings to a single switch. The policy contains the WWN of one or more switches, and the first one online becomes the primary FCS. The primary FCS can make changes and propagate fabric-wide parameters if an FCS policy is active. These parameters include the zoning database and security policies database.

As with other ACL policies, reviewing and adequately maintaining the FCS policy regularly is essential.

15.3.3 Device Connection Control (DCC) Policy

The DCC policy restricts devices through WWN from attaching to an Fibre Channel port. The DCC policy set comprises the DCC policies defined for each Fibre Channel port. The policy specifies the Fibre Channel port and one or more WWNs allowed to connect to that port. Not every Fibre Channel port needs a DCC policy; only the ports in the active policy set enforce access control. A port in the active DCC policy allows only the specified WWNs to connect and log in to the fabric. All other WWNs fail authentication when attempting to connect, which results in the corresponding F_Port being disabled due to a security violation.

Use the DCC policy in environments where strict control of fabric members is required. Since the DCC policy can prevent devices from participating in a fabric, it is essential to review and adequately maintain the DCC policy regularly.

15.3.4 Policy Database Distribution

Brocade Fabric OS provides a mechanism for controlling the distribution of the security policy database on a per-switch basis. Switches can use individually configured policies or a fabric-wide distributed policy on each platform. They can accept or reject a policy distributed from another switch. A fabric-wide distribution policy can be defined for SCC and DCC with support for strict, tolerant, and absent modes. The following modes enforce whether the SCC and DCC policy must be consistent throughout the fabric.

- **Strict mode:** All updated and new policies of the type specified (SCC, DCC, or both) must be distributed to all switches in the fabric, and all switches must accept the policy distribution.
- **Tolerant mode:** All updated and new policies of the type specified (SCC, DCC, or both) are distributed to all switches in the fabric, but the policy does not need to be accepted.
- **Absent mode:** Updated and new policies of the type specified (SCC, DCC, or both) are not automatically distributed to other switches in the fabric; policies can be manually distributed.

Together, the policy distribution and fabric-wide consistency settings provide a range of control over the security policies, from no control to strict control.

Refer to the *Brocade Fiber Channel Security Best Practices* for a detailed discussion of SAN security concepts and issues.

15.3.5 Authentication Protocols

Brocade Fabric OS supports Fibre Channel Authentication Protocols (FCAPs) and Diffie-Hellman Challenge Handshake Authentication Protocols (DH-CHAPs) on E_Ports and F_Ports. Authentication protocols provide additional security during link initialization by ensuring that only the desired device or device type connects to a given port.

15.4 Principle of Least Privilege (PoLP)

Starting with Fabric OS v10.x, it follows PoLP. The root account has been removed; no processes run as root. Fabric OS has been architected into a compartmentalization model, and all services and commands are granted only the minimum access necessary to perform the specific task. The new Fabric OS architecture drastically decreases the attack surface and potential impact of a security breach.

Fabric OS 10.x has implemented enhanced malware and ransomware detection. The user is alerted to contact support if detected, and the switch status is transitioned to marginal.

15.5 Secure SAN Management

User account and role management are vital to secure SAN management, with strong policies for accounts and passwords in combination with separation of duties and assigned privileges on a need-to basis only.

The following list outlines best practices for secure SAN management:

- Only secure protocols for connecting to Brocade platforms (SSH, HTTPS, and SNMPv3)
- Use a centralized account and password management system, such as Federated Authentication or AAA (Authentication, Authorization, and Accounting), like LDAP
- Each user must have their unique account with the proper role and privileges
- Change default passwords on *all* default accounts, and do not use the default accounts
- Create and enforce password policies (strength, history, expiration, and lockout)

15.5.1 Role-Based Access Control (RBAC)

One way to limit access to a fabric is through user roles. Brocade Fabric OS has predefined roles, each authorizing a subset of CLI commands. Predefined roles are called RBAC and are associated with user login credentials. RBAC aligns users with their function and authority. Users are granted specific privileges based on an organization’s security model to enforce the separation of duties. A role could be read-only, allowing users to view information without modifying or deleting it. A role can grant full admin privileges at the opposite end of the spectrum. Other roles fall in between and can be customized for specific functions, such as an operator or a security administrator.

15.6 Securing Management Interfaces

Management interfaces are a vulnerable point in any IT infrastructure; therefore, protecting them should always be a high priority and reasonably straightforward.

The following list outlines the measures to protect the management interfaces:

- Use a separate VLAN (or private VLAN) for the management network
- Use secure protocols to access management interfaces (SSH, HTTPS, and SNMPv3)
- Disable non-secure protocols (telnet, HTTP, and SNMPv1)
- Limit the entry points for management with an IP Filter policy and use an FCS policy if necessary.

Since malicious insiders can be a threat, protect management interfaces using a separate VLAN and subnet to isolate the management network. An isolated network limits access to SAN administrators, making access from other networks difficult. Use secure protocols to encrypt communications from management workstations to Brocade platforms. Encrypted protocols include SSH, HTTPS, and SNMPv3; disable the equivalent unsecured protocols. Starting with Fabric OS v10.x, unsecured protocols such as telnet, HTTP, and SNMPv1 are no longer available or supported.

15.6.1 IP Filter

The IPfilter policy restricts access to the IP management ports. It is a set of rules applied to the management interfaces, functioning as a packet-filtering firewall. According to the IPfilter rules, it will permit or deny (drop) ingress traffic. Only the IP addresses or subnets listed in the IPfilter rules can connect to specified TCP/UDP ports, if specified. IPfilter rule construction varies.

Brocade Fabric OS supports multiple IPfilter definitions. IPv4 and IPv6 IPfilter policies may be separate policies, or, starting with Brocade Fabric OS 10.0, the IPfilter policy can be unified for both IPv4 and IPv6. You cannot specify IPv6 addresses in an IPv4 policy or IPv4 addresses in an IPv6 policy; however, both IPv4 and IPv6 can be specified in a unified policy. Multiple filter policies can be defined for each type. For a unified IP Filter policy, there is only one active policy permitted. For IPv4 and IPv6 policies, one of each type can be active.

The IPfilter policy should only allow secure protocols. For stronger security in environments requiring strict access control, the source addresses or subnets for which SAN administration is performed should be the only permitted IP addresses. As with other ACL policies, regularly reviewing and adequately maintaining the IP Filter policy is crucial.

Chapter 16: Automation

16.1 Overview and Purpose

Most IT administrators have first-hand experience managing the growing complexity of enterprise infrastructure. According to a report from the Enterprise Strategy Group, “The cost and complexity of protecting and storing data is increasing, and IT leaders are responding with attempts to better optimize and automate storage—but better tools are needed.”

Broadcom is uniquely positioned to spot and understand the impact of automation, helping organizations get more from their SAN infrastructure. Broadcom offers a combination of SAN automation with RESTful APIs and a SAN management platform to help organizations drive greater efficiency. Automation and efficiency are accomplished through a variety of means:

- Brocade Automation multilayer architecture
- RESTful API support on platforms and management tools
- Broadcom Ansible management framework eliminates repetitive tasks, simplifies management, and orchestrates infrastructure

16.2 Motivation to Automate

The following are five reasons why organizations should embrace SAN automation:

- Reducing human error and streamlining operational processes have never been more crucial. As organizations move to digitize and adapt to new workloads, data availability, processing time, and agility in provisioning on-demand applications become the business’s lifeblood. These new digitized workloads demand a more efficient and expedient infrastructure management approach, leaving no room for human error. As a result, storage administrators need to be freed from repetitive manual tasks such as configuration management, reporting, documenting inventory, and troubleshooting. Instead, IT organizations need SAN automation to help them automate and orchestrate repetitive tasks, significantly improve efficiency, and decrease the risk of operational mistakes.
- Demand for more accurate and more frequent infrastructure reports is on the rise. It is not just IT managers who crave information about storage performance, utilization, and forecasting; business stakeholders are also asking for and expecting this data on demand. No one wants to wait for a slot when storage administrators can allocate time to produce a report. This information should be available as frequently as business demands dictate—all at the click of a button. Automation provides this kind of responsiveness that traditional manual storage management processes cannot deliver. Still, it can also be customized so that all stakeholders get more accurate data aligned to their responsibility.
- SAN configuration management must be streamlined. With more enterprise applications demanding access to more data and virtual machines, deploying and configuring servers, storage, and the network has become more time-consuming and complex than ever. By streamlining SAN operations through automation, application provisioning workflows are simplified across hypervisor, network, and storage, delivering agility and responsiveness to meet dynamic business demands.
- IT service delivery is not always as responsive as the business demands. As organizations increasingly rely on world-class IT services for proactive, agile business decision-making, they must identify and eliminate bottlenecks to IT service delivery. These enhanced IT services must be delivered without hiring more storage administrators or boosting SAN-related CapEx spending. SAN automation is the only viable option to drive increased agility and closely align IT services with fast-changing business needs.
- Consistent configuration validation is a must. As enterprises diversify their general IT and storage architectures, manual configuration changes occur more frequently. SAN automation ensures the validation of consistent configuration parameters across the different SAN fabrics to facilitate troubleshooting of frequent alerts without reliance on manual intervention.

Broadcom automation solutions leverage RESTful APIs to facilitate solutions architecture, share best practices, and get to production faster.

16.3 Overview of the REST API

The Brocade Fabric OS REST API is a programmable web service interface that manages Brocade platforms across a SAN. This API uses standard HTTPS methods to perform Create, Read, Update, and Delete (CRUD) operations on the fabric configuration data. It provides an interface for provisioning, status, and validation operations using the YANG data model described in the YANG 1.1 RFC, but not the data store managed with NETCONF. An Apache webserver embedded in Fabric OS is used to serve the API.

The RESTful API approach lets you think of a network device as a webserver. Automation can send and receive transactions to or from a network device by using standard web-based tools just as it would send transactions to and from a website. Transactions of this nature mean that they happen over a secure socket using HTTP rules to handle the exchange. The data appears in XML or JSON depending on the RESTful API services implemented on the networking device.

To interact with a SAN (or other) device, you need to consult its RESTful API reference to learn, among other things, what uniform resource identifiers (URIs) you need to use. (Simply put, URIs are identifiers that can be used as part of a web address.) According to the documentation, the URI for accessing a list of zones in the active configuration is as follows: `GET <base_URI>/running/zoning/defined-configuration/`

The model used to represent state and configuration information is expressed in a modeling language called Yang. Yang describes the structure of the different elements inside the model and describes whether each element is read-only or read-write. It describes the type of data that the element can hold, such as string or integer. It shows the relationship among various elements, the other nested elements they contain, their peer elements, and the parent elements that contain them. Here is a segment of the description of a zone in Yang:

```
list zone {
key "zone-name"; description
"List of the members in the zone. The members can be identified only as a WWN, domain, index, or zone
alias.";
leaf zone-name {
type zoning-name-type; description "The zone name.";
}
leaf zone-type {
type zone-type-type; description
"The zone type. Not that target zone types cannot be created or modified (only deleted).";
}
container member-entry { description
"The zone member."; leaf-list entry-name {
type zone-member-type; min-elements 1; description
"List of the members in the zone. The members can be identified only as a WWN, domain, index, or zone
alias.";
}

leaf-list principal-entry-name {
when "../..zone-type=1 or ../../zone-type=2"; type zone-member-type; min-elements 1; description
"List of the principal members in the peer zone. The members can be identified only as a WWN, domain,
index, or zone alias.";
}
}
}
```

Ordinarily, more information goes into a Yang module, such as revisioning and governance information; this listing omits them for brevity. Thus, the Yang description is complete, but it is also wordy. Although this precision is necessary when interacting with the model programmatically, it is sometimes helpful to get a global view of the abstraction provided by the model to see how the data is structured.

An open-source tool called pyang can parse the Yang model and produce a tree representing the model's elements. The listing includes information about each element, whether it is read-only or read-write, a list, optional, or nested. Here is the representation of the zoning model in tree form:

```
module: brocade-zone
+--rw brocade-zone
+--rw defined-configuration
| +--rw cfg* [cfg-name]
| | +--rw cfg-namezoning-name-type
|  |  +--rw member-zone
| | +--rw zone-name*zoning-name-type
| +--rw zone* [zone-name]
| | +--rw zone-namezoning-name-type
| | +--rw zone-type?zone-type-type
|  |  +--rw member-entry
| | +--rw entry-name*zone-member-type
| | +--rw principal-entry-name*zone-member-type
| +--rw alias* [alias-name]
| +--rw alias-namezoning-name-type
| +--rw member-entry
| +--rw alias-entry-name*union
+--rw effective-configuration
+--rw cfg-name?zoning-name-type
+--rw checksum?string
+--rw cfg-action?uint8
+--rw default-zone-access?uint8
+--ro db-max?uint32
+--ro db-avail?uint32
+--ro db-committed?uint32
+--ro db-transaction?uint32
+--ro transaction-token?uint32
+--ro db-chassis-wide-committed?uint32
+--ro enabled-zone* [zone-name]
+--ro zone-namezoning-name-type
+--ro zone-type?zone-type-type
+--ro member-entry
+--ro entry-name*union
+--ro principal-entry-name*union
```

16.4 Simple Automation Example

The `<base_URI>` is `http://<our device IP address>/rest` in this example. Begin by creating a login session with a switch in the fabric by executing the following command, which you type as a single line:

```
curl -X POST -v -u admin:password http://10.18.254.37/rest/login
```

where:

- `curl` is the command's name
- `-X POST` specifies the POST HTTP method (instead of GET)
- `-v` specifies verbose output to access the authorization string in the header of the response used in the next step
- `-u admin:password` specifies the credentials to use

The last parameter is the uniform resource identifier (URI) for curl to use to log in. (The URI value is described in the RESTful API reference. This command establishes the session used for the following commands. The following is a trace of its execution:

```
*  Trying 10.18.254.37...
* Connected to 10.18.254.37 (10.18.254.37) port 80
(#0)
* Server auth using Basic with user 'admin' POST /rest/login HTTP/1.1
>  Host: 10.18.254.37
Authorization: Basic YWRtaW46cGFzc3dvcmQ= User-Agent: curl/7.47.0
Accept: */* >
< HTTP/1.1 200 OK
< Date: Wed, 31 Jan 2018 16:01:24 GMT
<  Server: Apache
< Authorization: Custom_Basic YWRtaW46eHh4OjNkYTllZmM3NzMxYjk4OGU2ODg1YzZkMGRjNWJlM
zMyNjBhZDYxZThkOWQ2MWMxNzNiMGVlMjU3YmM2OTcyYjA=
< Cache-Control: no-cache
< X-Frame-Options: DENY
< Content-Secure-Policy: default-src 'self'
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Connection: close
< Transfer-Encoding: chunked
< Content-Type: application/yang-data+xml
```

Next, you perform a GET of the URI to return the current configuration using the Custom Basic value returned from the login for authentication:

```
curl -v -H "Authorization: Custom_Basic YWRtaW46eHh4OjNkYTllZmM3NzMxYjk4OGU2ODg1YzZkMGRjNWJlM
zMyNjBhZDYxZThkOWQ2MWMxNzNiMGVlMjU3YmM2OTcyYjA="
http://10.18.254.37/rest/running/zoning/defined-configuration
```

By default, `curl` uses the GET method, so you do not need to specify it. `-H "Authorization: Custom_Basic YWR...jA="` is the authentication and session identifying string returned in the previous command. `-H` places the string into the GET request header as seen in the following trace:

```
*  Trying 10.18.254.37...
* Connected to 10.18.254.37 (10.18.254.37) port 80
(#0)
GET /rest/running/zoning/defined-configuration HTTP/1.1 Host: 10.18.254.37
User-Agent: curl/7.47.0 Accept: */*
Authorization: Custom_Basic YWRtaW46eHh4OjNkYTllZmM3NzMxYjk4OGU2ODg1YzZkMGRjNWJlM
zMyNjBhZDYxZThkOWQ2MWMxNzNiMGVlMjU3YmM2OTcyYjA=
>
< HTTP/1.1 200 OK
< Date: Wed, 31 Jan 2018 16:09:39 GMT
<  Server: Apache
< Cache-Control: no-cache
< X-Frame-Options: DENY
< Content-Secure-Policy: default-src 'self'
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Connection: close
< Transfer-Encoding: chunked
< Content-Type: application/yang-data+xml <
<?xml version="1.0"?>
<Response>
<defined-configuration>
<cfg>
<cfg-name>CFG_FABRIC_A</cfg-name> <member-zone> <zone-name>CLUSTER1</zone-name>
```

```
<zone-name>Z_AIXHOST_FCS2_VMAX01_SN1234_9F0 </zone-name>
...
<alias> <alias-name>esx66_5d3d00</alias-name> <member-entry> <alias-entry-
name>10:00:8c:7c:ff:5d:3d:00 </alias-entry-name>
</member-entry>
</alias>
</defined-configuration>
</Response>
* Closing connection 0
```

The results appear as an XML data segment structured according to the Yang model's description, so it is crucial to have access to that model along with the RESTful API manual. The models can be found on GitHub as a repository among the Broadcom repositories at http://github.com/brocade/yang. The RESTful API manual can be retrieved from the Broadcom website. Having retrieved the zoning information from the fabric, you should close the session using the CLI command (the results are omitted to save space):

```
curl -v -H "Authorization: Custom_Basic YWRtaW46eHh4OjNkYTllZmM3NzMxYjk4OGU2ODg1YzZkMGRjNWJlM
zMyNjBhZDYxZThkOWQ2MWMxNzNiMGVlMjU3YmM2OTcyYjA=" http://10.18.254.37/rest/logout
```

In Fabric OS version 8.2.2 or later, the REST API session-less operation allows you to provide authentication credentials directly for each GET request. Essentially, the Fabric OS REST API session-less operation completes the login, GET operation, and logout as one complete request. You can use only basic authentication formats for REST API session-less operation, including plain text or Base64.

The following example shows a GET request using plain-text authentication:

```
curl -u admin:password http://10.155.2.190/rest/running/brocade-media/media-rdp>"
```

16.5 Ansible as an Alternative

The previous section shows an example of an approach that uses a procedural methodology. The workflow starts at the beginning, executes a series of steps, and then terminates. Most traditional programs work this way.

Ansible takes a declarative approach. Rather than provide sequential steps, Ansible describes each host in an inventory. The description appears in a document called a playbook. For example, Ansible describes a host state where the application is already installed rather than providing steps to install a particular application. When you run the playbook, Ansible takes no action if the application is already installed. If the application is not installed, Ansible calls installation routines to bring the host into the desired state without requiring the administrator to write specific steps.

In the realm of storage networks, using declarative language means that you can describe switches and fabrics where, for example, a zone is already configured with the proper hosts and storage arrays. When you run the Ansible playbook, those zones are defined as needed, and the hosts and storage arrays are added if necessary.

With some other declarative automation utilities, installing an agent on each managed host is necessary. This agent retrieves the commands from a command center and runs them on the localhost. Ansible is unique in that it does not require agents. Ansible establishes a secure shell session to a proxy and sends it a small Python script to make switch state changes. The script performs the necessary operations using the switch API and removes itself from the host.

You need two different skill sets to implement an Ansible solution successfully. First, you must understand the most common playbook operations. These operations are coded and installed for use by the playbooks. As vendors announce support for Ansible, they also provide script libraries for the most common tasks. Suppose there is a required task, but it is not available in the official Ansible distribution. In that case, the open-source community might provide code for that task in publicly available repositories.

Second, you must understand your business needs to provide ongoing playbook development. The person who maintains the playbooks does not need to be a programmer and does not need to know how remote system operations occur. That person needs to know only the desired outcomes and should be able to construct playbooks in YAML, the markup language used by Ansible. The following is an example of an Ansible playbook:

```
---
hosts: edgeSwitches vars_files:
-../fos_passwords.yml gather_facts: False tasks:

name: run fos commands brocade_fos_command:

switch_login: "{{switch_admin_account}}" switch_password: "{{switch_password}}" switch_address:
"{{switch_ip_address}}" command_set:
- command: alicreate "SampleAlias1", "10:23:45:67:76:54:32:10"
```

The three dashes at the beginning are part of the YAML specification. The `hosts` section identifies automation target switches. You can keep sensitive information in a separate file, as demonstrated by the `fos_passwords.yml` line. The name of a variable in double braces, such as `{{switch_password}}`, indicates variable substitution. The variable file specified in `vars_files` tells where to find external variables.

16.6 SANnav REST API

The SANnav Management Portal and SANnav Global View REST APIs provide end-to-end visibility into enterprise SANs. These tools detect, analyze, and take action based on SAN behavior and performance, enabling administrators to identify and fully remediate problems quickly. Storage administrators can troubleshoot across the storage fabric in as little as 30 seconds, capabilities unprecedented with any other shared storage infrastructure architecture.

The SANnav Management Portal REST APIs provide functionality that complements the Fabric OS REST APIs. The SANnav REST API feature details are in the *Brocade SANnav Management Portal REST API and Northbound Streaming Reference Manual*. Currently there are no REST interfaces provided by the SANnav Global View REST API.

16.7 Conclusion

SAN automation is a critical element in IT modernization and digital transformation. It helps organizations handle storage-related processes more efficiently without hiring more administrators or adding to the storage infrastructure CapEx budget. SAN automation is a high-leverage approach to turning network storage into a strategic asset. Broadcom's commitment to SAN automation and long-standing leadership in storage fabrics and technical innovation make it an ideal candidate for your IT infrastructure automation strategy.

Appendix A: Optical Cables

Table 3: FC Standard Supported Distances in Meters Based on Cable Type and Data Rates

Speed Name	OM1 Link Distance 62.5-µm Core and 200 MHz*km	OM2 Link Distance 50-µm Core and 500 MHz*km	OM3 Link Distance 50-µm Core and 2000 MHz*km	OM4/5 Link Distance 50-µm Core and 4700 MHz*km	OS1 Link Distance 9-µm Core and ~Infinite MHz*km
8G Fibre Channel	21	50	150	190	10,000
10G Fibre Channel	33	82	300	550	10,000
16G Fibre Channel	15	35	100	125	10,000
32G Fibre Channel	—	20	70	100	10,000
64G Fibre Channel	—	20	70	100	10,000
128G Fibre Channel	—	—	60	100	10,000

Table 4: LWL Optics Support (SFP+)

Transceiver Data Rate (Gb/s)	Distance (km)
8	10, 25
10	10
16	10, 25
32	10, 25
64	10, 25
128	10, 20

Appendix B: Fabric Details

This appendix provides example checklists and tables that you can use to identify dominant factors, including facilities that will have an impact on the SAN design.

Table 5: Current Fabrics

SAN/Fabric	No. of Switches	Type of Switches	Total Ports	Domains	No. of Servers	No. of Storage Devices	Location	Notes
Fabric 1								
Fabric 2								
Fabric 3								
Fabric 4								
Fabric 5								

Table 6: Individual Fabric Details

SAN/Fabric	Domain Number	Serial Number	Model	Speed	WWN	IP Addresses	Brocade Fabric OS Version	Notes
Switch 1								
Switch 2								
Switch 3								
Switch 4								
Switch 5								

Table 7: Device Details

Servers and Storage	Vendor	Model	WWN	Alias	Zone	OS Version	Application	Fabric/ Switches	Notes
Server 1									
Server 2									
Server 3									
Storage 1									
Storage 2									
Storage 3									

Table 8: Metrics and Impact on SAN Design and Performance

Metric	Source	Impact
Servers in the SAN	Estimate/Brocade SAN Health	Normal operations
Host Level Mirroring	Estimate	Distance, ISL congestion, traffic levels
Clusters (MSFT, HACMP, NetApp) Average number of nodes Workload level	Estimate Estimate: High/Med/Low	In-band heartbeat, frame congestion, host fan-in, traffic isolation
Virtualization: VIO Server No. of servers Consolidation ratio	Estimate Estimate Estimate	Frame congestion, edge traffic increase per port, server fan-in on target ports, device latencies
Virtualization: VMware No. of VMware servers Consolidated ratio? Shared VMFS? DRS? RDM? I/O intensive?	Estimate Estimate Yes/No Yes (%)/No Yes (%)/No High/Med/Low Yes/No	Frame congestion, device latencies, and SCSI2 reservations

Table 9: Consolidated SAN Snapshot

SAN Requirements Data (Complete for Each SAN)	
Fabric Information	
Target number of user ports per fabric	
Target number of total ports per fabric	
Target number of switches per fabric (number of switches/switch type, total switches)	
Number of fabrics	
Number of sites in environment	
Topology (core-edge, ring, mesh, other)	
Maximum hop count	
Expected growth rate (port count)	
Fabric licenses	
SAN Device Information	
Number/types of hosts and OS platforms	
Number/types of storage devices	
Number/types of tapes	
Number/types of HBAs	
Other devices (VTL/deduplication appliance)	
Total number of SAN devices per fabric	
Customer requirement for failover/redundancy, reliability of SAN (multipathing software utilized)	
SAN Requirements Data (Complete for Each SAN)	

Table 9: Consolidated SAN Snapshot (Continued)

Application Details	
SAN Application (Storage Consolidation, Backup and Restore, Business Continuance)	
Fabric management application(s)	
Performance	
Maximum latency (ms)	
Targeted ISL oversubscription ratio (3:1, 7:1, 15:1, other)	

Table 10: Application-Specific Details

Backup/Restore Infrastructure		
Servers		
System	OS Version, Patch Level	HBA Driver Version
Server 1/HBA		
Server 2/HBA		
Server 3/HBA		
Backup Software		
Vendor	Version	Patch
FC Switch		
Vendor	Model	Firmware
Brocade		
Storage		
Vendor	Model	Firmware
Array 1		
Array 2		
Tape Library		
Vendor	Model	Firmware
Library		

NOTE: Keep a similar table for each application.

Table 11: Quantitative Analysis: Radar Maps

SAN/Storage Admin Concerns	**Rank (1 is Low, 10 is High)**	**Notes**
ISL utilization	8	Is traffic balanced across ISLs during peaks?
Switch outage	1	Have there been switch outages? If so, what was the cause?
Zoning policy	6	Is the zoning policy defined?
Number of switches in the fabric	10	Is the current number of switches a concern for manageability?
Scalability	6	How many devices, ports, domains, hops, routes, switches, servers, storage… can be supported?
Redundancy	10	Is the existing SAN redundant for supporting a phased migration or firmware update?
Server: high availability	10	Does the cluster software fail over reliably?

Table 11: Quantitative Analysis: Radar Maps (Continued)

Storage: high availability	10	Do the LUNs fail over reliably?
Available disk pool	6	Is there a sufficient disk pool to support additional apps?
Management tools for SAN	4	Are the right tools used for SAN management?
Application response	7	Have there been any instances of slow application response but no outage?

Appendix C: References

C.1 Compatibility, Scalability, and Target Path

- Broadcom Customer Support Portal (CSP)
- Brocade SAN Scalability Guidelines: Brocade Fabric OS
- Brocade Fabric OS Target Path Selection Guide

C.2 Brocade SAN Health

- www.broadcom.com/sanhealth

C.3 Brocade Bookshelf

- NVMe over Fibre Channel for Dummies
- Networking Next-Gen Storage for Dummies
- SAN Automation for Dummies

C.4 Other

- The SNIA Dictionary

Revision History

53-1004781-08; November 18, 2025

- Added 128 Gb/s LWL Optics Support (SFP+) to Table 4.
- Grammatical fixes.

53-1004781-07; November 18, 2025

- Updated for Gen 8 and Fabric OS 10.0

53-1004781-06; May 10, 2024

- Updated long-distance measurements in Predeployment Cabling and Optics Validation.

53-1004781-05; August 15, 2023

- Adjusted cover page content

53-1004781-04; July 2023

- Added updated for Brocade 7850.
- Reviewed, updated, and edited the design guide in its entirety.

53-1004781-03; January 2023

- Added updates for Fabric OS 9.2.
- Removed Extension (Extension now has its own similar document).
- Reviewed, updated, and edited the design guide in its entirety.

53-1004781-03; May 2022

- Added updates for Fabric OS 9.1.
- Reviewed, updated, and edited the design guide in its entirety.

53-1004781-02; September 1, 2020

- Added updates for Fabric OS 9.0 and Gen 7.
- Added the “Automation” chapter.

53-1004781-01; November 23, 2016

Initial release.