

# Dell EMC PowerScale: Network Design Considerations

## Abstract

This white paper explains design considerations of the Dell EMC™ PowerScale™ external network to ensure maximum performance and an optimal user experience.

February 2021

## Revisions

Date	Description
March 2017	Initial rough draft
July 2017	Updated after several reviews and posted online
November 2017	<p>Updated after additional feedback. Updated title from “Isilon Advanced Networking Fundamentals” to “Isilon Network Design Considerations.” Updated the following sections with additional details:</p> <ul style="list-style-type: none"> <li>• Link Aggregation</li> <li>• Jumbo Frames</li> <li>• Latency</li> <li>• ICMP &amp; MTU</li> <li>• New sections added:</li> <li>• MTU Framesize Overhead</li> <li>• Ethernet Frame</li> <li>• Network Troubleshooting</li> </ul>
December 2017	<p>Added link to Network Stack Tuning spreadsheet</p> <p>Added Multi-Chassis Link Aggregation</p>
January 2018	<p>Removed switch-specific configuration steps with a note for contacting manufacturer</p> <p>Updated section title for Confirming Transmitted MTUs</p> <p>Added OneFS commands for checking and modifying MTU</p> <p>Updated Jumbo Frames section</p>
May 2018	Updated equation for Bandwidth Delay Product
August 2018	<p>Added the following sections:</p> <ul style="list-style-type: none"> <li>• SyncIQ Considerations</li> <li>• SmartConnect Considerations</li> <li>• Access Zones Best Practices</li> </ul>
August 2018	Minor updates based on feedback and added ‘Source-Based Routing Considerations’
September 2018	Updated links
November 2018	Added section ‘Source-Based Routing & DNS’
April 2019	Updated for OneFS 8.2: Added SmartConnect Multi-SSIP
June 2019	Updated SmartConnect Multi-SSIP section based on feedback.
July 2019	Corrected errors

Date	Description
August 2019	Updated Ethernet flow control section
January 2020	Updated to include 25 GbE as front-end NIC option.
April 2020	Added 'DNS and time-to-live' section and added 'SmartConnect Zone Aliases as opposed to CNAMEs' section.
May 2020	Added 'S3' section under 'Protocols and SmartConnect allocation methods', updated 'Isilon' branding to 'PowerScale', and added 'IPMI' section.
June 2020	Added 'QoS' and 'Software-Defined Networking' sections. Updated the 'NFSv4' section with Kerberos and updated the 'IP Address quantification' section.
July 2020	Added 'SmartConnect service name' section.
August 2020	Added 'Isilon 6 <sup>th</sup> generation 1 GbE interfaces' and 'VLAN and interface MTU' sections. Updated 'IPMI' section.
September 2020	Updated 'DNS delegation best practices' and 'SmartConnect in isolated network environments' sections.
February 2021	Updated 'IPMI' section.

## Acknowledgements

Author: Aqib Kazi

The information in this publication is provided "as is." Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

This document may contain certain words that are not consistent with Dell's current language guidelines. Dell plans to update the document over subsequent future releases to revise these words accordingly.

This document may contain language from third party content that is not under Dell's control and is not consistent with Dell's current guidelines for Dell's own content. When such third party content is updated by the relevant third parties, this document will be revised accordingly

Copyright © 2021 Dell Inc. or its subsidiaries. All Rights Reserved. Dell Technologies, Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners. [2/2/2021] [Technical White Paper] [H16463.19]

# Table of contents

Revisions.....	2
Acknowledgements.....	3
Table of contents .....	4
Executive summary.....	8
Note to readers .....	8
1 Network architecture design.....	9
1.1 General network architecture considerations .....	9
1.2 Triangle looped topology .....	10
1.3 Link aggregation .....	11
1.3.1 Multi-chassis link aggregation .....	12
2 Latency, bandwidth, and throughput .....	13
2.1 Latency .....	13
2.2 Bandwidth and throughput.....	14
2.2.1 Bandwidth delay product .....	14
2.3 PowerScale network stack tuning.....	15
3 Ethernet flow control.....	17
3.1 Checking for pause frames.....	17
3.1.1 4 <sup>th</sup> and 5 <sup>th</sup> generation Isilon nodes .....	18
3.1.2 6 <sup>th</sup> generation Isilon nodes .....	18
4 SyncIQ considerations .....	19
4.1 SyncIQ disaster recovery with SmartConnect.....	19
4.2 Replication traffic over dedicated WAN links.....	19
5 Quality of Service (QoS).....	20
6 Software-Defined Networking.....	21
7 PowerScale OneFS ports .....	22
8 SmartConnect considerations .....	23
8.1 SmartConnect network hierarchy .....	23
8.2 Load balancing .....	24
8.3 Static or dynamic IP address allocation .....	25
8.4 Dynamic failover .....	25
8.4.1 Dynamic failover examples.....	26
8.5 Protocols and SmartConnect allocation methods .....	28
8.5.1 SMB .....	28
8.5.2 NFS.....	29

8.5.3 HDFS .....	29
8.5.4 S3 .....	29
8.5.5 Suggested zones by protocol .....	30
8.6 IP address quantification .....	31
8.7 SmartConnect service name .....	32
8.8 SmartConnect node suspension .....	33
8.9 SmartConnect and Reverse DNS.....	33
8.10 DNS delegation best practices .....	34
8.10.1 Delegate to address (A) records, not to IP addresses .....	34
8.10.2 SmartConnect zone aliases as opposed to CNAMEs.....	34
8.10.3 One name server record for each SmartConnect zone name or alias .....	34
8.10.4 Multiple DNS resolvers in a groupnet.....	35
8.11 SmartConnect in isolated network environments .....	36
8.12 SmartConnect DNS, subnet, and pool design.....	36
8.12.1 SmartConnect zone naming.....	37
8.12.2 SmartConnect with multiple node pools or types .....	37
8.13 Where the SmartConnect Service IP (SSIP) runs (pre OneFS 8.2).....	38
8.14 SmartConnect Multi-SSIP .....	40
8.14.1 Configuring OneFS for SmartConnect Multi-SSIP .....	41
8.14.2 Configuring a DNS server for SmartConnect multi-SSIP .....	42
8.14.3 SSIP node assignment.....	43
8.15 DNS and time-to-live .....	44
8.15.1 Microsoft Windows DNS.....	44
8.15.2 BIND DNS .....	44
8.16 Other SmartConnect considerations .....	45
<b>9 Ethernet, MTU, and IP overhead.....</b>	<b>46</b>
9.1 Ethernet packet.....	46
9.2 Ethernet payload.....	47
9.3 Jumbo frames .....	47
9.4 IP packet overhead.....	48
9.4.1 Example 1: Standard 1500-byte payload – IPv4/TCP.....	48
9.4.2 Example 2: Jumbo 9000-byte payload – IPv4/TCP.....	49
9.4.3 Example 3: Standard 1500-byte payload – IPv4/TCP/Linux timestamp .....	49
9.4.4 Example 4: Jumbo 9000-byte payload – IPv4/TCP/Linux timestamp .....	49
9.5 Data payload to Ethernet frame efficiency .....	50
9.6 ICMP and MTU with OneFS .....	50

9.7	OneFS MTU commands .....	51
9.8	VLAN and interface MTU .....	51
9.9	Confirming transmitted MTU .....	51
10	Access Zones best practices .....	52
10.1	System Zone .....	52
10.2	Root Based Path .....	52
11	Source-Based Routing considerations .....	54
11.1	Source-Based Routing and DNS .....	55
12	Isilon 6 <sup>th</sup> generation 1 GbE interfaces .....	56
13	Intelligent Platform Management Interface .....	57
13.1	Configuring IPMI .....	58
13.2	IPMI SoL on PowerScale nodes .....	59
13.2.1	Configure serial devices .....	59
13.2.2	iDRAC SoL permission .....	59
13.3	Accessing IPMI .....	60
13.4	Troubleshooting IPMI .....	60
14	IPv6 .....	61
14.1	Why IPv6? .....	61
14.1.1	Security .....	61
14.1.2	Efficiency .....	61
14.1.3	Multicast .....	61
14.1.4	Quality of Service .....	61
14.2	IPv6 addressing .....	62
14.3	IPv6 header .....	63
14.4	IPv6 to IPv4 translation .....	63
14.5	Configuring OneFS for IPv6 .....	64
15	Network troubleshooting .....	65
15.1	Netstat .....	65
15.1.1	Netstat .....	65
15.1.2	netstat -s -p tcp .....	66
15.1.3	netstat -i .....	67
15.1.4	netstat -m .....	68
15.2	InsightIQ external network errors .....	68
15.3	DNS .....	70
A	Supported network optics and transceivers .....	72
B	Technical support and resources .....	73

B.1 Related resources.....73

## Executive summary

This document provides design considerations for understanding, configuring, and troubleshooting PowerScale Scale-Out NAS external networking. In a Scale-Out NAS environment, the overall network architecture must be configured to maximize the user experience. Many factors contribute to overall network performance. This document examines network architecture design and best practices including factors such as Latency, Flow Control, ICMP, MTU, jumbo frames, congestion, TCP/IP parameters, and IPv6.

## Note to readers

It is important to understand that the network design considerations stated in this document are based on general network design and are provided as guidance to PowerScale administrators. As these are considerations, all of these may not apply to each workload. It is important to understand each consideration and confirm if it pertains to a specific environment.

Each network is unique, not only from a design perspective but also from a requirements and workloads perspective. Before making any changes based on the guidance in this document, it is important to discuss modifications with the Network Engineering team. Additionally, as a customary requirement for any major IT implementation, changes should first be tested in a lab environment that closely mimics the workloads of the live network.



# 1 Network architecture design

The architecture design is the core foundation of a reliable and highly available network, considering capacity and bandwidth. Layered on top of the basic foundation are the many applications running on a campus network with each requiring specific features and considerations.

For the following sections, it is important to understand the differences between distribution and access switches. Typically, distribution switches perform L2/L3 connectivity while access switches are strictly L2. Figure 1 provides the representation for each.

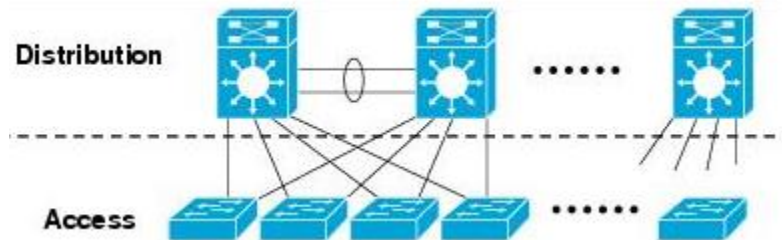


Figure 1 Distribution and Access Switches

## 1.1 General network architecture considerations

Designing a network is unique to the requirements of each enterprise data center. There is certainly not a “one size fits all” design and not a single “good network design.” When approaching network design, it is important to use principles as a leading factor, coupled with the enterprise requirements. The requirements must include current and future application consumption, providing the guiding factor in major decisions. Network design is based on many concepts; the following are considerations and principles to guide the process:

- **Single Points of Failure:** Ensure the network design has layers of redundancy. Dependence on a single device or link relates to a loss of resources or outages. The enterprise requirements consider risk and budget, guiding the level of redundancy. Redundancy should be implemented through backup paths and load sharing. If a primary link fails, traffic uses a backup path. Load sharing creates two or more paths to the same endpoint and shares the network load. When designing access to PowerScale nodes, it is important to assume links and hardware will fail, ensuring access to the nodes survives those failures.
- **Application and Protocol Traffic:** Understanding the application data flow from clients to the PowerScale cluster across the network allows for resources to be allocated accordingly while minimizing latency and hops along this flow.
- **Available Bandwidth:** As traffic traverses the different layers of the network, the available bandwidth should not be significantly different. Compare this available bandwidth with the workflow requirements.
- **Minimizing Latency:** Ensuring latency is minimal from the client endpoints to the PowerScale nodes maximizes performance and efficiency. Several steps can be taken to minimize latency, but latency should be considered throughout network design.
- **Prune VLANs:** It is important to limit VLANs to areas where they are applicable. Pruning unneeded VLANs is also good practice. If unneeded VLANs are trunked further down the network, this imposes additional strain on endpoints and switches. Broadcasts are propagated across the VLAN and impact clients.

- **VLAN Hopping:** VLAN hopping has two methods, switch spoofing and double tagging. Switch spoofing is when a host imitates the behavior of a trunking switch, allowing access to other VLANs. Double tagging is a method where each packet contains two VLAN tags, with the assigned or correct VLAN tag is empty and the second as the VLAN where access is not permitted. It is recommended to assign the native VLAN to an ID that is not in use. Otherwise tag the native VLAN to avoid VLAN hopping, allowing a device to access a VLAN it normally would not have access. Additionally, only allow trunk ports between trusted devices and assign access VLANs on ports that are different from the default VLAN.

## 1.2 Triangle looped topology

This section provides best practices for Layer 2 Access network design. Although many network architectures may meet enterprise requirements, this document takes a closer look at what is commonly referred to as the Triangle Looped Access Topology, which is the most widely implemented architecture in enterprise data centers.

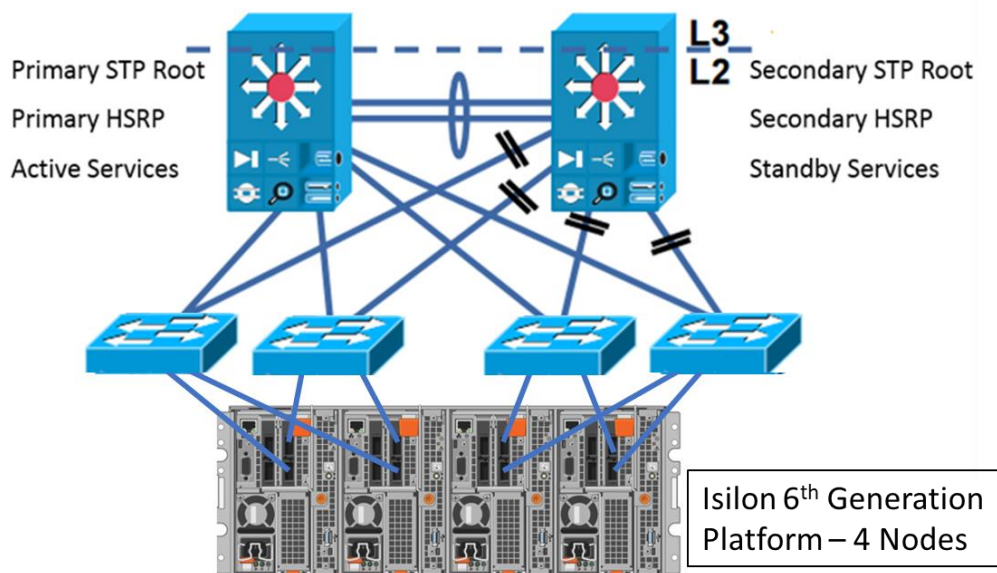


Figure 2 Triangle Looped Access Topology

The Looped Design Model extends VLANs between the aggregation switches, thus creating the looped topology. To prevent actual loops, Spanning Tree is implemented, using Rapid PVST+ or MST. For each path, a redundant path also exists, which is blocking until the primary path is not available. Access layer uplinks may be used to load balance VLANs. A key point to consider with the Looped Access Topology is the utilization of the inter-switch link between the Distribution switches. The utilization must be monitored closely as this is used to reach active services.

The Looped Triangle Access Topology supports VLAN extension and L2 adjacency across the Access layer. Through the use of STP and dual homing, the Looped Triangle is extremely resilient. Stateful services are supported at the aggregation layer and quick convergence with 802.1W/S.

Utilizing the Triangle Looped Topology allows for multiple Access Switches to interface with the external network of the PowerScale Scale-Out NAS environment. Each PowerScale node within a cluster is part of a distributed architecture which allows each node to have similar properties regarding data availability and management.

## 1.3 Link aggregation

In the context of the IEEE 802.1AX standard, link aggregation provides methods to combine multiple Ethernet interfaces, forming a single link layer interface, specific to a switch or server. Therefore, link aggregation is implemented between a single switch and a PowerScale node, not across PowerScale nodes.

Implementing link aggregation is neither mandatory nor is it necessary, rather it is based on workload requirements and is recommended if a transparent failover or switch port redundancy is required.

Link aggregation assumes all links are full duplex, point to point, and at the same data rate, providing graceful recovery from link failures. If a link fails, traffic is automatically sent to the next available link without disruption.

It is imperative to understand that link aggregation is not a substitute for a higher bandwidth link. Although link aggregation combines multiple interfaces, applying it to multiply bandwidth by the number of interfaces for a single session is incorrect. Link aggregation distributes traffic across links. However, a single session only utilizes a single physical link to ensure packets are delivered in order without duplication of frames.

As part of the IEEE 802.1AX standard, the Frame Distributor does not specify a distribution algorithm across aggregated links but enforces that frames must be sent in order without duplication. Frame order is maintained by ensuring that all frames of a given session are transmitted on a single link in the order that they are generated by the client. The mandate does not allow for additions or modifications to the MAC frame, buffering, or processing to re-order frames by the Frame Distributor or Collector.

Thus, the bandwidth for a single client is not increased, but the aggregate bandwidth of all clients increases in an active/active configuration. The aggregate bandwidth is realized when carrying multiple simultaneous sessions and may not provide a linear multiple of each link's data rate, as each individual session utilizes a single link.

Another factor to consider is depending on the workload, certain protocols may or may not benefit from link aggregation. Stateful protocols, such as NFSv4 and SMBv2 benefit from link aggregation as a failover mechanism. On the contrary, SMBv3 Multichannel automatically detects multiple links, utilizing each for maximum throughput and link resilience.

Table 1 Link Aggregation

Link Aggregation Advantages	Link Aggregation Limitations
Higher aggregate bandwidth for multiple sessions. A single session is confined to a single link.	Provides resiliency for interface and cabling failures, but not for switch failures.
Link resiliency	Bandwidth for a single session is not improved as a single link is used for each session.
Ease of management with a single IP address	Depending on the workload, each protocol has varying limitations and advantages of Link Aggregation
Load balancing	

OneFS supports round-robin, failover, load-balance, and LACP link aggregation methods. In previous releases, FEC was also listed as an option. However, FEC was simply the naming convention for load-balance. In OneFS 8.2, load-balance replaces the FEC option.

### 1.3.1 Multi-chassis link aggregation

As discussed in the previous section, the IEEE 802.1AX standard does not define Link Aggregation between multiple switches and a PowerScale node. However, many vendors provide this functionality through proprietary features. Multiple switches are connected with an Inter-Switch link or other proprietary cable and communicate via a proprietary protocol forming a virtual switch. A virtual switch is perceived as a single switch to a PowerScale node, with links terminating on a single switch. The ability to have link aggregation split with multiple chassis provides network redundancy if a single chassis were to fail.

Each vendor has a proprietary implementation of Multi-Chassis Link Aggregation, but externally the virtual switch created is compliant with the IEEE 802.1AX standard.

It is important to recognize that regarding bandwidth, the concepts discussed for single switch Link Aggregation still apply to Multi-Chassis Link Aggregation. Additionally, as the multiple switches form a single virtual switch, it is important to understand what happens if the switch hosting the control plane fails. Those effects vary by the vendor's implementation but will impact the network redundancy gained through Multi-Chassis Link Aggregation.

## 2 Latency, bandwidth, and throughput

Maximizing overall network performance is dependent on several factors. However, the three biggest factors contributing to end-to-end performance are latency, throughput, and bandwidth. This section focuses on these factors to maximize the PowerScale user experience.

### 2.1 Latency

Latency in a packet-switched network is defined as the time from when a source endpoint sends a packet to when it is received by the destination endpoint. Round trip latency, sometimes referred to as round-trip delay, is the amount of time for a packet to be sent from the source endpoint to the destination endpoint and returned from the destination to the source endpoint.

Minimal latency in any transaction is imperative for several reasons. IP endpoints, switches, and routers operate optimally without network delays. Minimal latency between clients and a PowerScale node ensures performance is not impacted. As latency increases between two endpoints, this may lead to several issues that degrade performance heavily, depending on the application.

In order to minimize latency, it is important to measure it accurately between the endpoints. For assessing PowerScale nodes, this is measured from the clients to a specified node. The measurement could use the IP of a specific node or the SmartConnect hostname. After configuration changes are applied that impact latency, it is important to confirm the latency has indeed decreased. When attempting to minimize latency, consider the following points:

- **Hops:** Minimizing hops required between endpoints decreases latency. The implication is not to drag cables across a campus, but the goal is to confirm if any unnecessary hops could be avoided. Minimizing hops applies at the physical level with the number of switches between the endpoints but also applies logically to network protocols and algorithms.
- **ASICs:** When thinking about network hops it also important to consider the ASICs within a switch. If a packet enters through one ASIC and exits through the other, latency could increase. If at all possible, it is recommended to keep traffic as part of the same ASIC to minimize latency.
- **Network Congestion:** NFS v3, NFSv4 and SMB employ the TCP protocol. For reliability and throughput, TCP uses windowing to adapt to varying network congestion. At peak traffic, congestion control is triggered, dropping packets, and leading TCP to utilize smaller windows. In turn, throughput could decrease, and overall latency may increase. Minimizing network congestion ensures it does not impact latency. It is important to architect networks that are resilient to congestion.
- **Routing:** Packets that pass through a router may induce additional latency. Depending on the router configuration, packets are checked for a match against defined rules, in some cases requiring packet header modification.
- **MTU Mismatch:** Depending on the MTU size configuration of each hop between two endpoints, an MTU mismatch may exist. Therefore, packets must be split to conform to upstream links, creating additional CPU overhead on routers and NICs, creating higher processing times, and leading to additional latency.
- **Firewalls:** Firewalls provide protection by filtering through packets against set rules for additional steps. The filtering process consumes time and could create further latency. Processing times are heavily dependent upon the number of rules in place. It is good measure to ensure outdated rules are removed to minimize processing times.

## 2.2 Bandwidth and throughput

Understanding the difference between throughput and bandwidth are important for network troubleshooting. Although these terms are conflated at times, they are actually both unique. Bandwidth is the theoretical maximum speed a specific medium can deliver if all factors are perfect without any form of interference. Throughput is the actual speed realized in a real-world scenario, given interference and other environmental factors such as configuration, contention, and congestion.

The difference between these terms is important when troubleshooting. If a PowerScale node supports 40 GbE, it does not necessarily mean the throughput is 40 Gb/s. The actual throughput between a client and a PowerScale node is dependent on all of the factors between the two endpoints and may be measured with a variety of tools.

During the design phase of a data center network, it is important to ensure bandwidth is available throughout the hierarchy, eliminating bottlenecks and ensuring consistent bandwidth. The bandwidth from the Access Switches to the PowerScale nodes should be a ratio of what is available back to the distribution and core switches. For example, if a PowerScale cluster of 12 nodes has all 40 GbE connectivity to access switches, the link from the core to distribution to access should be able to handle the throughput from the access switches. Ideally, the link from the core to distribution to access should support roughly a bandwidth of 480 Gb (12 nodes \* 40 GbE).

### 2.2.1 Bandwidth delay product

Bandwidth Delay Product (BDP) is calculated to find the amount of data a network link is capable of, in bytes, which can be transmitted on a network link at a given time. The keyword is transmitted, meaning the data is not yet acknowledged. BDP takes into consideration the bandwidth of the data link and the latency on that link, in terms of a round-trip delay.

The amount of data that can be transmitted across a link is vital to understanding Transmission Control Protocol (TCP) performance. Achieving maximum TCP throughput requires that data must be sent in quantities large enough before waiting for a confirmation message from the receiver, which acknowledges the successful receipt of data. The successful receipt of the data is part of the TCP connection flow. The diagram below explains the steps of a TCP connection and where BDP is applicable:

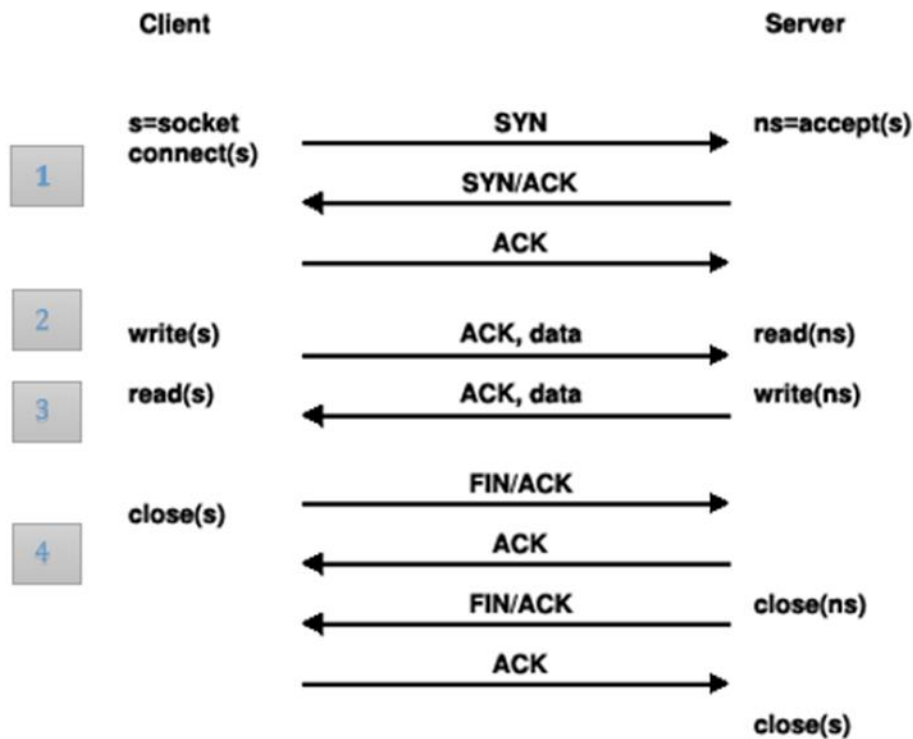


Figure 3 Transmission Control Protocol Message Flow

In the diagram above, four states are highlighted during a TCP connection. The following summarizes each state:

1. TCP Handshake – Establishes the TCP connection through an SYN, SYN/ACK, ACK
2. Data transmitted to the server. BDP is the maximum amount of data that can be sent at this step.
3. Data acknowledged by Server
4. TCP Connection Close Sequence – Socket closure is initiated by either side

Once the BDP rate is calculated, the TCP stack is tuned for the maximum throughput, which is discussed in the next section. The BDP is calculated by multiplying the bandwidth of the network link (bits/second) by the round-trip time (seconds).

For example, a link with a bandwidth of 1 Gigabit per second and a 1 millisecond round trip time, would be calculated as:

Bandwidth \* RTT = 1 Gigabit per second \* 1 millisecond =  
 1,000,000,000 bits per second \* 0.001 seconds = 1,000,000 bits = 0.125 MB

Thus, 0.125 MB may be sent per TCP message to the server.

## 2.3 PowerScale network stack tuning

Once the BDP is calculated and understood, these findings can be applied to modifying the TCP stack on the PowerScale cluster. All PowerScale clusters do not require TCP stack tuning. Only alter the TCP stack for a needed workflow improvement. The majority of PowerScale environments do not need TCP tuning. Before applying any TCP changes, ensure the network is clean and reliable by performing basic checks for excessive retransmits, duplicate or fragmented packets, and broken pipes.



PowerScale OneFS is built on FreeBSD. A PowerScale cluster is composed of nodes with a distributed architecture, and each node provides external network connectivity. Adapting the TCP stack to bandwidth, latency, and MTU requires tuning to ensure the cluster provides optimal throughput.

In the previous section, BDP was explained in depth and how it is the amount of data that can be sent across a single TCP message flow. Although the link supports the BDP that is calculated, the OneFS system buffer must be able to hold the full BDP. Otherwise, TCP transmission failures may occur. If the buffer does not accept all of the data of a single BDP, the acknowledgment is not sent, creating a delay, and the workload performance is degraded.

The OneFS network stack must be tuned to ensure on inbound, the full BDP is accepted, and on outbound, it must be retained for a possible retransmission. Prior to modifying the TCP stack, it is important to measure the current I/O performance and then again after implementing changes. As discussed earlier in this document, the tuning below is only guidance and should be tested in a lab environment before modifying a production network.

The spreadsheet below provides the necessary TCP stack changes based on the bandwidth, latency, and MTU. The changes below must be implemented in the order below and all together on all nodes. Modifying only some variables could lead to unknown results. After making changes, it is important to measure performance again.

**Note:** The snippet below is only for representation. It is imperative to input the calculated bandwidth, latency, and MTU specific to each environment.

		Enter values below		
Available bandwidth (Gb/s)		40		
Latency (ms)		5		
MTU		1500		
Bandwidth delay product (MB)				
		Sys Controls		
		Rece		
				Default value
102400	kern.ipc.maxsockbuf	104857600		2097152
25600	net.inet.tcp.recvspace	26214400		131072
25600	net.inet.tcp.sendspace	26214400		131072
51200	net.inet.tcp.recvbuf_max	52428800		262144
51200	net.inet.tcp.sendbuf_max	52428800		262144
16	net.inet.tcp.sendbuf_inc	16384		8192
32	net.inet.tcp.recvbuf_inc	32768		16384
	net.inet.tcp.rfc1323	1		1
	net.inet.tcp.sack.enable	1		1
	net.inet.tcp.mssdflt	1448		512
	net.inet.tcp.path_mtu_discovery	1		1

Figure 4 PowerScale TCP network stack tuning

Download the PowerScale Network Stack Tuning spreadsheet at the following link:

<https://dellmc.com/resources/en-us/asset/technical-guides-support-information/h164888-isilon-onefs-network-stack-tuning.xlsm>



## 3 Ethernet flow control

Under certain conditions, packets sent from the source to the destination can overwhelm the destination endpoint. The destination is not able to process all packets at the rate that they are sent, leading to retransmits or dropped packets. Most scenarios have a fast source endpoint and a slower destination endpoint; this could be due to processing power or several source endpoints interacting with a single destination. Flow control is implemented to manage the rate of data transfer between these IP endpoints, providing an option for the destination to control the data rate, and ensuring the destination is capable of processing all of the packets from the source.

The IEEE 802.3x standard defines an Ethernet Flow Control mechanism at the data link layer. It specifies a **pause** flow control mechanism through MAC Control frames in full-duplex link segments. For flow control to be successfully implemented, it must be configured throughout the network hops that the source and destination endpoints communicate through. Otherwise, the pause flow control frames are not recognized and are dropped.

By default, PowerScale OneFS listens for pause frames but does not transmit them, meaning it is only applicable when a PowerScale node is the source. In the default behavior, OneFS recognizes pause frames from the destination. However, pause frames may be enabled for transmit, depending on the NIC.

Most network devices today do not send pause frames, but certain devices still send them.

### 3.1 Checking for pause frames

If the network or cluster performance does not seem optimal, it is easy to check for pause frames on a PowerScale cluster.

If pause frames are reported, it is important to discuss these findings with the network engineering team before making any changes. As mentioned above, changes must be implemented across the network, ensuring all devices recognize a pause frame. Contact the switch manufacturer's support teams or account representative for specific steps and caveats for implementing flow control before proceeding.

### 3.1.1 4<sup>th</sup> and 5<sup>th</sup> generation Isilon nodes

On a 4<sup>th</sup> or 5<sup>th</sup> generation Isilon cluster, check for pause frames received by executing the following command from the shell:

```
isi_for_array -a <cluster name> sysctl dev | grep pause
```

**Check for any values greater than zero.** In the example, below, the cluster has not received any pause frames. If values greater than zero are printed consistently, flow control should be considered.

```
tme-sandbox-1# isi_for_array -a tme-sandbox sysctl dev | grep pause
tme-sandbox-3: dev.bxe.0.rx_pause_frames: 0
tme-sandbox-3: dev.bxe.0.rx_constant_pause_events: 0
tme-sandbox-3: dev.bxe.0.tx_pause_frames: 0
tme-sandbox-3: dev.bxe.1.rx_pause_frames: 0
tme-sandbox-3: dev.bxe.1.rx_constant_pause_events: 0
tme-sandbox-3: dev.bxe.1.tx_pause_frames: 0
tme-sandbox-2: dev.bxe.0.rx_pause_frames: 0
tme-sandbox-2: dev.bxe.0.rx_constant_pause_events: 0
tme-sandbox-2: dev.bxe.0.tx_pause_frames: 0
tme-sandbox-2: dev.bxe.1.rx_pause_frames: 0
tme-sandbox-2: dev.bxe.1.rx_constant_pause_events: 0
tme-sandbox-2: dev.bxe.1.tx_pause_frames: 0
tme-sandbox-1: dev.bxe.0.rx_pause_frames: 0
tme-sandbox-1: dev.bxe.0.rx_constant_pause_events: 0
tme-sandbox-1: dev.bxe.0.tx_pause_frames: 0
tme-sandbox-1: dev.bxe.1.rx_pause_frames: 0
tme-sandbox-1: dev.bxe.1.rx_constant_pause_events: 0
tme-sandbox-1: dev.bxe.1.tx_pause_frames: 0
```

Figure 5 Checking for pause frames

### 3.1.2 6<sup>th</sup> generation Isilon nodes

For 6<sup>th</sup> generation Isilon nodes with ix NICs, check for pause frames with the following commands:

```
infPerf-1# sysctl -d dev.ix.0.mac_stats.xon_txd
dev.ix.0.mac_stats.xon_txd: Link XON Transmitted <<<Pause frame sent
infPerf-1# sysctl -d dev.ix.0.mac_stats.xon_rcvd
dev.ix.0.mac_stats.xon_rcvd: Link XON Received <<<Pause frame received
infPerf-1# sysctl -d dev.ix.0.mac_stats.xoff_txd
dev.ix.0.mac_stats.xoff_txd: Link XOFF Transmitted <<<Resume frame sent
infPerf-1# sysctl -d dev.ix.0.mac_stats.xoff_rcvd
dev.ix.0.mac_stats.xoff_rcvd: Link XOFF Received <<<Resume frame received
```

## 4 SyncIQ considerations

PowerScale SyncIQ provides asynchronous data replication for disaster recovery and business continuance, allowing failover and failback between clusters. It is configurable for either complete cluster replication or only for specific directories. Within a PowerScale cluster, all nodes can participate in replication. After an initial SyncIQ replication, only changed data blocks are copied minimizing network bandwidth and resource utilization on clusters.

This section provides considerations for SyncIQ pertaining to external network connectivity. For more information on SyncIQ, refer to the [PowerScale SyncIQ: Architecture, Configuration, and Considerations](#) white paper.

### 4.1 SyncIQ disaster recovery with SmartConnect

This section describes best practices for disaster recovery planning with OneFS SmartConnect.

Dedicated static SmartConnect zones are required for SyncIQ replication traffic. As with any static SmartConnect zone, the dedicated replication zone requires one IP address for each active logical interface. For example, in the case of two active physical interfaces, 10gige-1 and 10gige-2, requiring two IP addresses. However, if these are combined with link aggregation, interface 10gige-agg-1 only requires one IP address. Source-restrict all SyncIQ jobs to use the dedicated static SmartConnect zone on the source cluster and repeat the same on the target cluster.

By restricting SyncIQ replication jobs to a dedicated static SmartConnect Zone, replication traffic may be assigned to specific nodes, reducing the impact of SyncIQ jobs on user or client I/O. The replication traffic is directed without reconfiguring or modifying the interfaces participating in the SmartConnect zone.

For example, consider a data ingest cluster for a sports television network. The cluster must ingest large amounts of data recorded in 4K video format. The data must be active immediately, and the cluster must store the data for extended periods of time. The sports television network administrators want to keep data ingestion and data archiving separate, to maximize performance. The sports television network purchased two types of nodes: H500s for ingesting data, and A200s for the long-term archive. Due to the extensive size of the data set, SyncIQ jobs replicating the data to the disaster recovery site, have a significant amount of work to do on each pass. The front-end interfaces are saturated on the H500 nodes for either ingesting data or performing immediate data retrieval. The CPUs of those nodes must not be effected by the SyncIQ jobs. By using a separate static SmartConnect pool, the network administrators can force all SyncIQ traffic to leave only the A200 nodes and provide maximum throughput on the H500 nodes.

### 4.2 Replication traffic over dedicated WAN links

Depending on the network topology and configuration, in certain cases PowerScale SyncIQ data may be sent across a dedicated WAN link separated from client traffic. Under these circumstances, the recommended option is utilizing a different subnet on the PowerScale cluster for replication traffic, separated from the subnet for user data access.

## 5 Quality of Service (QoS)

As more applications compete for a shared link with limited throughput, ensuring Quality of Service (QoS) for application success is critical. Each application has varying QoS requirements to deliver not only service availability, but also an optimal client experience. Associating each application to an appropriate QoS marking, provides traffic policing, allowing packets to be prioritized as required across a shared medium, all while delivering an ideal client experience.

QoS may be implemented through different methods. However, the most common is through a Differentiated Services Code Point (DSCP), specifying a value in the packet header that maps to an effort level for traffic.

PowerScale OneFS does not provide an option for tagging packets with a specified DSCP marking. As a best practice, configure the first hop ports on switches connected to PowerScale nodes to insert DSCP values. It is important to note that OneFS does retain headers for packets that already have a specified DSCP value.

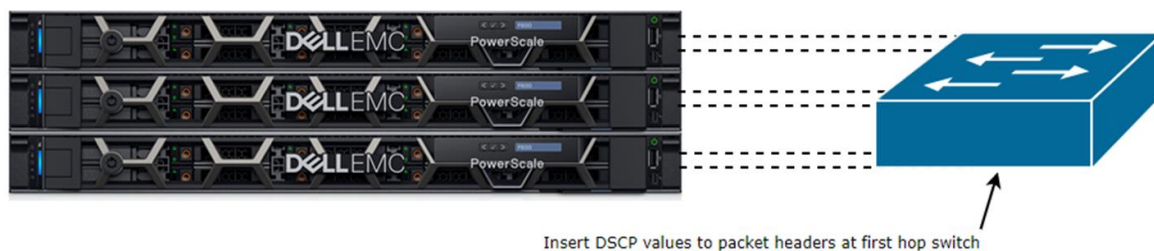


Figure 6 QoS and OneFS – Inserting DSCP values

## 6 Software-Defined Networking

Software-Defined Networking (SDN) provides automated policy-based management of network architecture. The management and administration are centralized by separating the control and data planes. SDN architectures include a controller functioning as a central point of management and automation. The controller is responsible for relaying information downstream to firewalls, routers, switches, and access points. On the contrary, the controller sends information upstream to applications and orchestration frameworks, all while presenting the SDN architecture as a single device.

Datacenters that have an SDN architecture and a PowerScale cluster must have traditional access switches connected to PowerScale nodes, presenting a traditional network architecture to OneFS.

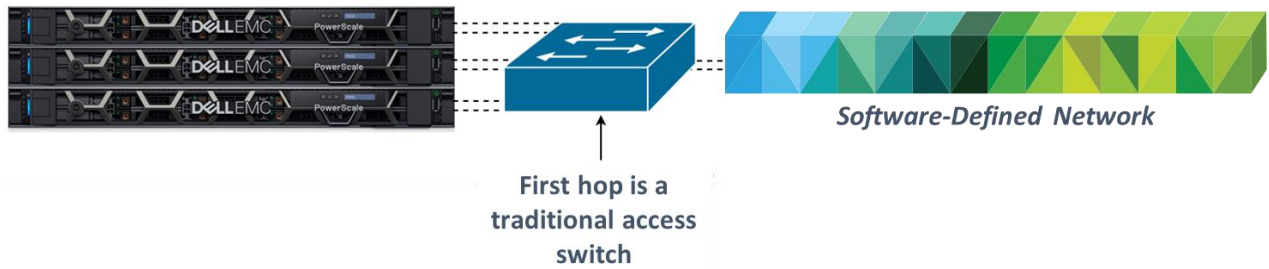


Figure 7 PowerScale and Software-Defined Networking

The SDN implementation of each vendor is unique and it is critical to understanding the scalability and limitations of a specific architecture. Some of the SDN implementations are based on open standards like OpenFlow, while other vendors use a mix of proprietary and open standards, and others use a completely proprietary implementation. Reviewing the limits of a specific implementation is essential to understanding how to maximize performance. If a PowerScale cluster is configured for use with SDN through a traditional access switch, consider the following:

- OneFS does not support VRFs and VXLANs. An intermediary solution is required for implementing VLAN to VXLAN mapping.
- Understand the control plane scalability of each SDN implementation and if it would impact OneFS.
- The MTU implementation for each vendor varies. Ensure consistent MTUs across all network hops.
- Each switch vendor provides a different set of SDN capabilities. Mapping differences is key to developing a data center architecture to include a PowerScale cluster while maximizing network performance.
- Not only is each vendor's capability unique when it comes to SDN. But, the scalability of each solution and cost varies significantly. The intersection of scalability and cost determines the architecture limits.
- As each SDN implementation varies, consider the impacts on the automation and policy-driven configuration, as this is one of the significant advantages of SDN. Additionally, consider the automation interactions with Isilon PAPI.

## 7 PowerScale OneFS ports

PowerScale OneFS uses a number of TCP and UDP ports, which are documented in the Security Configuration Guide available at the following link: <https://community.emc.com/docs/DOC-57599>

## 8 SmartConnect considerations

This section provides considerations for using the PowerScale SmartConnect load-balancing service. The general IP routing principles are the same with or without SmartConnect.

SmartConnect acts as a DNS delegation server to return IP addresses for SmartConnect zones, generally for load-balancing connections to the cluster. The IP traffic involved is a four-way transaction shown in Figure 8.



Figure 8 SmartConnect DNS delegation steps

In Figure 8, the arrows indicate the following steps:

1. **Blue arrow (step 1):** The client makes a DNS request for `sc-zone.domain.com` by sending a DNS request packet to the site DNS server.
2. **Green arrow (step 2):** The site DNS server has a delegation record for `sc-zone.domain.com` and sends a DNS request to the defined nameserver address in the delegation record, the SmartConnect service (SmartConnect Service IP Address).
3. **Orange arrow (step 3):** The cluster node hosting the SmartConnect Service IP (SSIP) for this zone receives the request, calculates the IP address to assign based on the configured connection policy for the pool in question (such as round robin), and sends a DNS response packet to the site DNS server.
4. **Red arrow (step 4):** The site DNS server sends the response back to the client.

### 8.1 SmartConnect network hierarchy

As SmartConnect subnets and pools are defined it is important to understand the SmartConnect hierarchy, as displayed in the following figure:

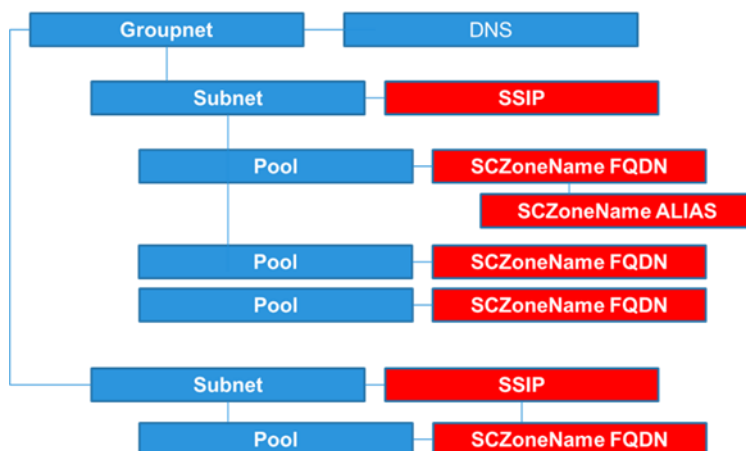


Figure 9 SmartConnect network hierarchy – OneFS releases prior to 8.2

Throughout the network design phase, for releases prior to OneFS 8.2, consider that a single SSIP is defined per subnet. However, under each subnet, pools are defined, and each pool will have a unique SmartConnect Zone Name. It is important to recognize that multiple pools lead to multiple SmartConnect Zones utilizing a



single SSIP. As shown in the diagram above, a DNS provider is defined per Groupnet, which is a feature in OneFS 8.0 and newer releases. In releases before 8.0, a DNS per Groupnet was not supported.

OneFS 8.2 introduces support for multiple SSIPs per subnet, as displayed in the following figure:

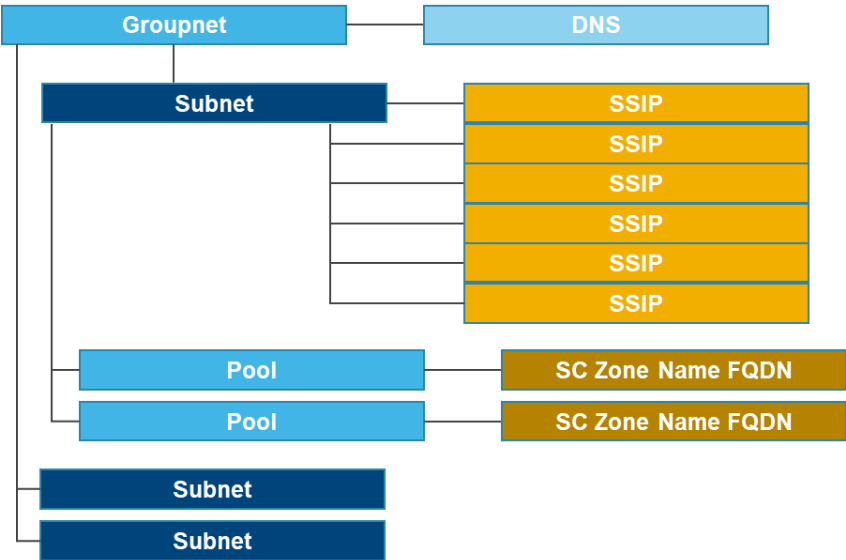


Figure 10 SmartConnect network hierarchy – OneFS release 8.2

For more information on SmartConnect multi-SSIP, refer to Section 8.14, SmartConnect Multi-SSIP.

8.2 Load balancing

SmartConnect load balances incoming network connections across SmartConnect Zones composed of nodes, network interfaces, and pools. The load balancing policies are Round Robin, Connection Count, CPU Utilization, and Network Throughput. The most common load balancing policies are Round Robin and Connection Count, but this may not apply to all workloads. It is important to understand whether the front-end connections are being evenly distributed, either in count or by bandwidth. Front-end connection distribution may be monitored with InsightIQ or the WebUI. It is important to understand how each Load Balancing Policy functions and testing it in a lab environment prior to a production roll-out, as each workload is unique. The table below lists suggested policies based on the workflow, but these are general suggestions, and may not always be applicable.

Generally speaking, starting with Round Robin is recommended for a new implementation or if the workload is not clearly defined. As the workload is further defined and based on the Round Robin experience, another policy can be tested in a lab environment.

Table 2 Suggested SmartConnect load balancing policies

Load Balancing Policy	Workload				
	General or Other	Few Clients with Extensive Usage	Many Persistent NFS & SMB Connections	Many Transitory Connections (HTTP, FTP)	NFS Automounts or UNC Paths
Round Robin	✓	✓	✓	✓	✓



Connection Count*	✓	✓		✓	✓
CPU Utilization*					
Network Throughput*					

\*Metrics are gathered every 5 seconds for CPU Utilization and every 10 seconds for Connection Count and Network Throughput. In cases where many connections are created at the same time, these metrics may not be accurate, creating an imbalance across nodes.

As discussed previously, the above policies mapping to workloads are general guidelines. Each environment is unique with distinct requirements. It is recommended to confirm the best load balancing policy in a lab environment which closely mimics the production environment.

## 8.3 Static or dynamic IP address allocation

After a groupnet and subnet are defined in OneFS, the next step is configuring an IP address pool and assigning interfaces to participate in this pool.

Once the IP address pool is defined, under the 'SmartConnect Advanced' Section, an 'Allocation Method' may be selected. By default, this option is grayed out as 'Static' if a SmartConnect Advanced license is not installed. If a SmartConnect Advanced license is installed, the default 'Allocation Method' is still 'Static', but 'Dynamic' may also be selected.

The Static Allocation Method assigns a single persistent IP address to each interface selected in the pool, leaving additional IP addresses in the pool unassigned if the number of IP addresses is greater than interfaces. The lowest IP address of the pool is assigned to the lowest Logical Node Number (LNN) from the selected interfaces, subsequently for the second lowest IP address and LNN. In the event a node or interface becomes unavailable, this IP address does not move to another node or interface. Additionally, when the node or interface becomes unavailable, it is removed from the SmartConnect Zone, and new connections will not be assigned to the node. Once the node is available again, SmartConnect adds it back into the zone and assigns new connections.

On the contrary, the Dynamic Allocation Method splits all available IP addresses in the pool across all selected interfaces. Under the Dynamic Allocation Method, OneFS attempts to assign the IP addresses evenly if at all possible, but if the interface to IP address ratio is not an integer value, a single interface may have more IP addresses than another.

## 8.4 Dynamic failover

Combined with the Dynamic Allocation Method, Dynamic Failover provides high-availability by transparently migrating IP addresses to another node when an interface is not available. If a node becomes unavailable, all of the IP addresses it was hosting are re-allocated across the new set of available nodes in accordance with the configured failover load balancing policy. The default IP address failover policy is round-robin, which evenly distributes IP addresses from the unavailable node across available nodes. As the IP address remains consistent, irrespective of which node it resides on, this results in a transparent failover to the client, providing seamless high availability.

The other available IP address failover policies are the same as the initial client connection balancing policies, i.e., connection count, throughput, or CPU usage. In most scenarios, round-robin is not only the best option,

but also the most common. However, the other failover policies are available for specific workflows. As mentioned previously, with the initial load balancing policy, test the IP failover policies in a lab environment to find the best option for a specific workflow.

### 8.4.1 Dynamic failover examples

In order to understand Dynamic Failover, the following examples illustrate how IP addresses move during a failover.

The examples below illustrate the concepts of how the IP address quantity impacts user experience during a failover, and these are the guidelines to use when determining IP address quantity.

#### 8.4.1.1 Dynamic Allocation with 1 IP address per node

This example considers a four-node cluster with one network connection per node and one dynamic SmartConnect zone with only four IP addresses. One IP address will be assigned to each node, as shown in the following figure:

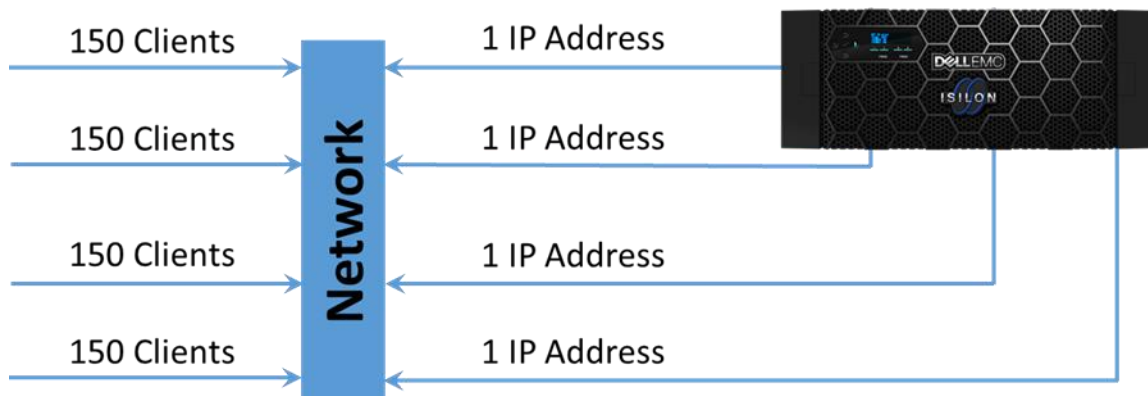


Figure 11 Dynamic Allocation: 4 node cluster with 1 IP address per node

In this scenario, 150 clients are actively connected to each node over NFS using a round-robin connection policy. Most NFSv3 mounted clients perform a nslookup only the first time that they mount, never performing another nslookup to check for an updated IP address. If the IP address changes, the NFSv3 clients have a stale mount and retain that IP address.

Suppose that one of the nodes fails, as shown in Figure 12.

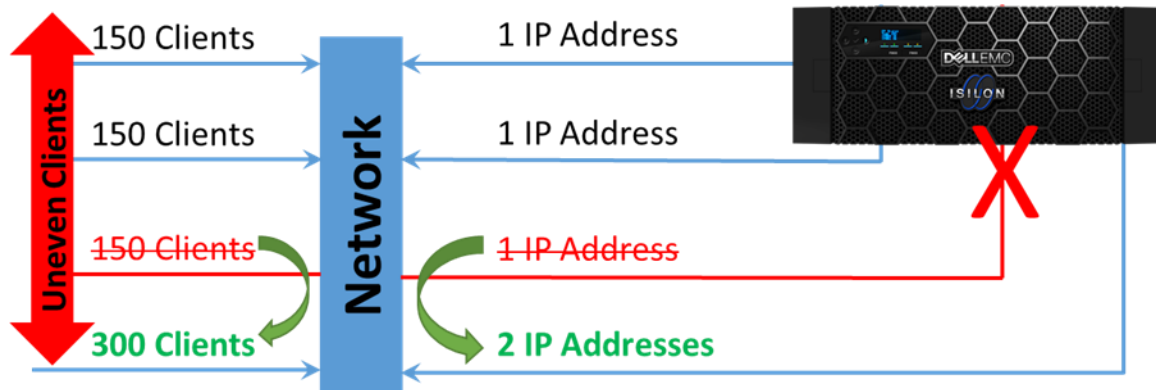


Figure 12 Dynamic Allocation: 4 node cluster with 1 IP address per node – 1 node offline

A SmartConnect Zone with Dynamic Allocation for IP addresses immediately hot-moves the one IP address on the failed node to one of the other three nodes in the cluster. It sends out a number of gratuitous address resolution protocol (ARP) requests to the connected switch, so that client I/O continues uninterrupted.

Although all four IP addresses are still online, two of them—and 300 clients—are now connected to one node. In practice, SmartConnect can fail only one IP to one other place, and one IP address and 150 clients are already connected to each of the other nodes. The failover process means that a failed node has just doubled the load on one of the three remaining nodes while not disrupting the other two nodes. Therefore, this process results in declining client performance, but not equally. The goal of any scale-out NAS solution must be consistency. To double the I/O on one node and not on another is inconsistent.

#### 8.4.1.2 Dynamic Allocation with 3 IP addresses per node

Dynamic SmartConnect zones require a greater number of IP addresses than the number of nodes at a minimum to handle failover behavior. In the example below, the formula used to calculate the number of IP addresses required is  $N*(N-1)$ , where 'N' is the number of nodes. The formula is used for illustration purposes only to demonstrate how IP addresses, and in turn, clients, move from one node to another, and how this could potentially lead to an imbalance across nodes. Every workflow and cluster is unique, and this formula is not applicable to every scenario.

This example considers the same four-node cluster as the previous example, but now following the rule of  $N*(N-1)$ . In this case,  $4*(4-1) = 12$ , equaling three IPs per node, as shown in Figure 13.

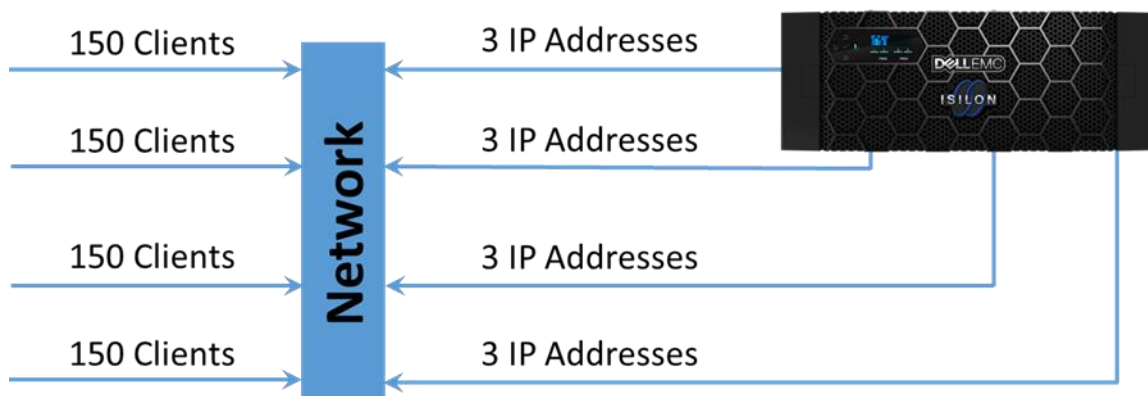


Figure 13 Dynamic Allocation: 4 node cluster with 3 IP addresses per node

When the same failure event as the previous example occurs, the three IP addresses are spread over all the other nodes in that SmartConnect zone. This failover results in each remaining node having 200 clients and four IP addresses. Although performance may degrade to a certain degree, it may not be as drastic as the failure in the first scenario, and the experience is consistent for all users, as shown in the following figure.

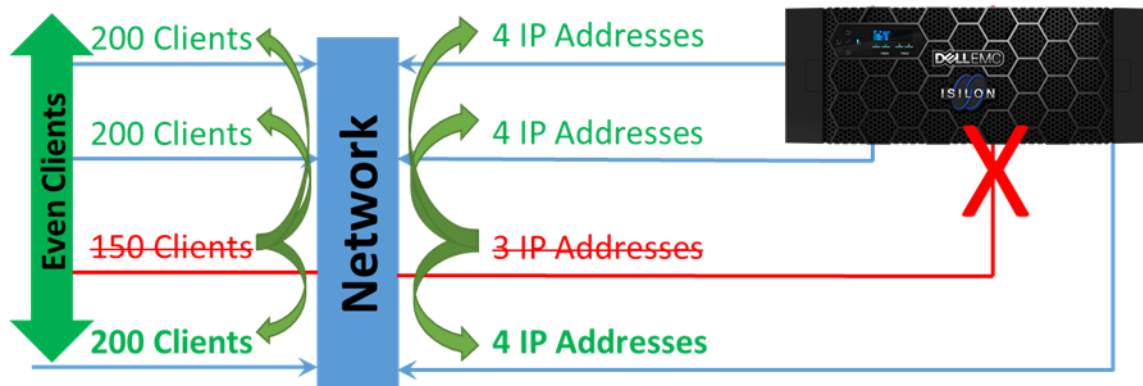


Figure 14 Dynamic Allocation: 4 node cluster with 3 IP addresses per node, 1 node offline

## 8.5 Protocols and SmartConnect allocation methods

A common concern during a PowerScale configuration is selecting between Static and Dynamic Allocation methods. The requirement for Dynamic Failover depends heavily on the protocol in use, workflow, and overall high-availability design requirements. Stateful versus stateless protocols combined with the allocation method, impact the failover experience. Certain workflows require minimal downtime, or the overarching IT requirements dictate IP address persistence. This section provides guidance on failover behavior based on the protocol.

Client access protocols are either stateful or stateless. Stateful protocols are defined by the client/server relationship having a session state for each open file. Failing over IP addresses to other nodes for these types of workflows means that the client assumes that the session state information was carried over. Session state information for each file is not shared among PowerScale nodes. On the contrary, stateless protocols are generally accepting of failover without session state information being maintained, except for locks.

**Note:** For static zones, ensure SmartConnect is configured with a time-to-live of zero. For more information, refer to Section 8.15, DNS and time-to-live.

### 8.5.1 SMB

Typically, SMB performs best in static zones. In certain workflows, SMB is preferred with Dynamic Allocation of IP addresses, because IP address consistency is required. It may not only be a workflow requirement but could also be an IT administrative dependence. SMB actually works well with Dynamic Allocation of IP addresses, but it is essential to understand the protocol limitations. SMB preserves complex state information per session on the server side. If a connection is lost and a new connection is established with dynamic failover to another node, the new node may not be able to continue the session where the previous one had left off. If the SMB workflow is primarily reads or is heavier on the read side, the impact of a dynamic failover will not be as drastic, as the client can re-open the file and continue reading.

Conversely, if an SMB workflow is primarily writes, the state information is lost, and the writes could be lost, possibly leading to file corruption. Hence, in most cases, static zones are suggested for SMB, but again it is

workflow dependent. Prior to a major implementation, it is recommended to test the workflow in a lab environment, understanding limitations and the best option for a specific workflow.

### 8.5.2 NFS

The NFSv2 and NFSv3 protocols are stateless, and in almost all cases, perform best with Dynamic Allocation of IP addresses. The client does not rely on writes unless commits have been acknowledged by the server, enabling NFS to failover dynamically from one node to another.

The NFSv4 protocol introduced state, making it a better fit for static zones in most cases, as it expects the server to maintain session state information. However, OneFS 8.0 introduced session-state information across multiple nodes for NFSv4, making dynamic pools the better option. Additionally, most mountd daemons currently still behave in a v3 manner, where if the IP address it's connected to becomes unavailable, this results in a stale mount. In this case, the client does not attempt a new nslookup and connect to a different node.

Another factor to consider for NFSv4 is if Kerberos authentication is configured. For Kerberos environments with NFSv4, static allocation is recommended. For non-Kerberos environments, dynamic allocation is recommended.

Again, as mentioned above, test the workflow in a lab environment to understand limitations and the best option for a specific workflow.

### 8.5.3 HDFS

The requirements for HDFS pools have been updated with the introduction of new OneFS features and as HDFS environments have evolved. During the design phases of HDFS pools, several factors must be considered. The use of static versus dynamic pools are impacted, by the following:

- Use of OneFS racks if needed
- Node Pools: is the cluster a single heterogeneous node type or do different Node Pools exist
- Availability of IP addresses

The factors above, coupled with the workflow requirements, determine the pool implementation. Please reference the HDFS Pool Usage and Assignments section in the [EMC PowerScale Best Practices Guide for Hadoop Data Storage](#) for additional details and considerations with HDFS pool implementations.

### 8.5.4 S3

OneFS 9.0 introduces support for Amazon's Simple Storage Service (S3) protocol. The S3 protocol is stateless. For most workflows, S3 performs optimally with Dynamic Allocation of IP addresses, ensuring seamless client failover to an available node in an event where the associated node becomes unavailable.

### 8.5.5 Suggested zones by protocol

The table below lists the suggested IP allocation strategies for SmartConnect Advanced by the protocol. As noted, these are suggested, and the actual zone type is dependent on the workflow requirements, as discussed above.

Table 3 Suggested protocols and zone types

Protocol	Protocol Category	<u>Suggested Zone Type</u>
NFSv2 (not supported in OneFS 7.2 and above)	Stateless	Dynamic
NFSv3	Stateless	Dynamic
NFSv4	Stateful	Dynamic or Static – Depending on mountd daemon, OneFS version, and Kerberos. Refer to the NFS section above.
SMBv1	Stateful	Dynamic or Static – Refer to SMB section above
SMBv2 / SMBv2.1	Stateful	
SMBv3 Multi-Channel	Stateful	
FTP	Stateful	Static
SFTP / SSH	Stateful	Static
HDFS	Stateful – Protocol is tolerant of failures	Refer to <a href="#">EMC PowerScale Best Practices Guide for Hadoop Data Storage</a>
S3	Stateless	Dynamic
HTTP / HTTPS	Stateless	Static
SyncIQ	Stateful	Static Required

## 8.6 IP address quantification

This section provides guidance for determining the number of IP addresses required for a new cluster implementation. The guidance provided below does not apply to all clusters and is provided as a reference for the process and considerations during a new cluster implementation.

During the process of implementing a new cluster and building the network topology, consider the following:

- Calculate the number of IP addresses that are needed based on future cluster size, not the initial cluster size.
- Do not share a subnet with other application servers. If more IP addresses are required, and the range is full, re-addressing an entire cluster and then moving it into a new VLAN is disruptive. These complications are prevented with proper planning.
- Static IP pools require one IP address for each logical interface that will be in the pool. Each node provides 2 interfaces for external networking. If Link Aggregation is not configured, this would require  $2*N$  IP addresses for a static pool.
- 1 IP address for each SmartConnect Service IP (SSIP)
- For optimal load-balancing, during a node failure, IP pools with the Dynamic Allocation Method require the number of IP addresses at a minimum of the node count and a maximum of the client count. For example, a 12-node SmartConnect zone and 50 clients, would have a minimum of 12 and a maximum of 50 IP addresses. In many larger configurations, defining an IP address per client is not feasible, and in those cases, the optimal number of IP addresses is workflow dependent and based on lab testing. In the previous examples,  $N*(N-1)$  is used to calculate the number of IP addresses, where  $N$  is the number of nodes that will participate in the pool. For larger clusters, this formula may not be feasible due to the sheer number of IP addresses. Determining the number of IP addresses within a Dynamic Allocation pool varies depending on the workflow, node count, and the estimated number of clients that would be in a failover event.
- If more than a single Access Zone is configured with IP pools using the Dynamic Allocation Method, examine if all the pools are required. Reducing the number of IP pools, will also reduce the number of IP addresses required.
- If a cluster has multiple Access Zones or IP pools, a lower number of IP addresses may be required. If so, consider reducing the total number of IP addresses. Generally, as more Access Zones and IP address pools are configured, fewer IP addresses are required.

In previous OneFS releases, a greater IP address quantity was recommended considering the typical cluster size and the workload a single node could handle during a failover. As nodes become unavailable, all the traffic hosted on that node is moved to another node with typically the same resources, which could lead to a degraded end-user experience. As PowerScale nodes are now in the 7<sup>th</sup> generation, this is no longer a concern. Each node does have limitations, and those must be considered when determining the number of IP addresses and failover events creating additional overhead. Additionally, as OneFS releases have progressed, so has the typical cluster size, making it difficult to maintain the  $N*(N-1)$  formula with larger clusters.

From a load-balancing perspective, for dynamic pools, it is ideal, although optional, that all the interfaces have the same number of IP addresses, whenever possible. It is important to note that in addition to the points above, consider the workflow and failover requirements set by IT administrators.



## 8.7 SmartConnect service name

The “SmartConnect service name” field is displayed when creating or modifying a subnet, as exhibited in Figure 15. The “Create subnet” dialog appears in the web interface by clicking “Cluster Management > Network Configuration” and then adding a new subnet under a specified groupnet. Alternatively, from the command line interface, this field appears when adding or modifying a subnet to a specified groupnet with the `--sc-service-name` option.

Figure 15 SmartConnect service name

The “SmartConnect service name” field is an optional field to answer nameserver (NS), Start of Authority (SOA), and other DNS queries. It specifies the domain name corresponding to the SmartConnect Service IP (SSIP) address, serving as the glue record in the DNS delegation tying the NS and the IP address. The DNS delegation to SmartConnect consists of 2 DNS records, as listed in Table 4.

Table 4 SmartConnect service name DNS records

DNS Record	DNS Field	DNS Value	OneFS Field
NS	Domain	cluster.company.com	SmartConnect Zone Name on the network pool ( <code>sc-dns-zone</code> in the CLI)
	Value	ns.cluster.company.com	SmartConnect Service Name on the subnet ( <code>sc-service-name</code> in the CLI)
	Description	This entry informs the DNS server the nameserver for the PowerScale cluster is this record at ns.cluster.company.com	
A/AAAA	Domain	ns.cluster.company.com	SmartConnect Service Name on the subnet ( <code>sc-service-name</code> in the CLI)
	Value	1.2.3.4	SmartConnect Service IP address on the subnet ( <code>sc-service-addr</code> in the CLI)
	Description	This entry informs the DNS server the nameserver for the PowerScale cluster can be contacted at the 1.2.3.4 IP address specified as the ‘A/AAAA Value’.	

**Note:** If a value is not provided for this field, SmartConnect re-uses the domain name from the nameserver and Start of Authority queries as the nameserver hostname. In the event the `sc-service-name` on the cluster is different than the record in the DNS Delegation, DNS resolution failures can occur. Due to this, it is highly advised to ensure these records are in sync.



## 8.8 SmartConnect node suspension

OneFS SmartConnect provides an option to administratively remove a node from a SmartConnect Zone during a planned outage. Planned outages could be hardware replacement or maintenance activity.

Once a node is suspended, SmartConnect prevents new client connections to the node. If the node is configured for Dynamic Allocation of IP addresses, IP addresses are not assigned to this node in a suspended state. Suspending a node ensures that client access remains consistent. After the node is suspended, client connections can be monitored and allowed to gradually drop-off before a reboot or power down.

A node is suspended from the OneFS CLI or web interface. From the OneFS CLI, the command is:

```
isi network pools --sc-suspend-node <groupnet.subnet.pool> <node ID>
```

Alternatively, from the web interface, click “Suspend Nodes” under the ‘Pool,’ as displayed in the following figure:

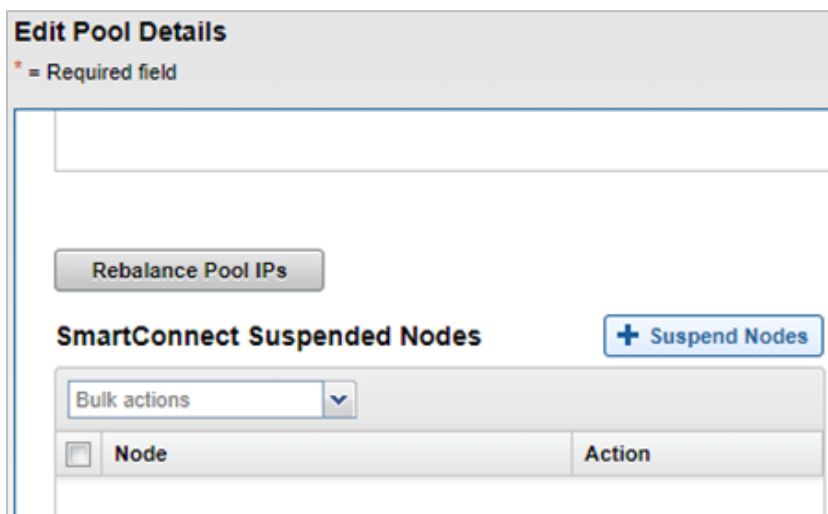


Figure 16 SmartConnect Node Suspension

After a node is suspended, new connections are not created. Prior to rebooting or shutting the node down, confirm all client connections have dropped by monitoring the web interface under the “Client Connections” tab from the “Cluster Overview” page. Also, clients may have to be manually booted from the node if they have static SMB connections with applications that maintain connections.

## 8.9 SmartConnect and Reverse DNS

In most cases, it is recommended that PowerScale SmartConnect Service IP addresses and SmartConnect Zone names, do not have reverse DNS entries, also known as pointer (PTR) records.

In certain environments where PTR records may be required, this results in the creation of many PTR entries, as PowerScale SmartConnect pools could have hundreds of IP addresses. In scenarios where PTR records are required, each time an additional IP address is added to a SmartConnect pool, DNS changes are necessary to keep the environment consistent.

Creating reverse DNS entries for the SmartConnect Service IP's Host [address, or A] record is acceptable if the SmartConnect Service IP is referenced only with an A record in one DNS domain.

## 8.10 DNS delegation best practices

This section describes DNS delegation best practices for PowerScale clusters.

### 8.10.1 Delegate to address (A) records, not to IP addresses

The SmartConnect service IP address on a PowerScale cluster, in most cases, should be registered in DNS as an address (A) record, also referred to as a 'Host Entry'. For example, the following SSIP (A) record would designate the SSIP record with a corresponding IP address:

```
cls01-ssip.foobar.com.      IN      A      192.168.255.10
```

In this case, the (A) record maps the URL, `cls01-ssip.foobar.com`, to a corresponding IP address of `192.168.255.10`. Delegating a SmartConnect zone to an (A) record simplifies failover and failback in business continuity, maintenance, and disaster recovery scenarios. In such cases, the change requirement remains minimal as only a single DNS record, the (A) record, would require an update.

All other SmartConnect zone delegations that are configured against the SSIP can be left alone as per the example:

```
cls01-smb.foobar.com.      IN      NS      cls01-ssip.foobar.com
cls01-nfs.foobar.com.      IN      NS      cls01-ssip.foobar.com
cls01-hdfs.foobar.com.      IN      NS      cls01-ssip.foobar.com
```

### 8.10.2 SmartConnect zone aliases as opposed to CNAMEs

A Canonical Name (CNAME) record is a DNS resource mapping one domain to another domain. CNAMEs are not recommended with OneFS, as it is not possible to discover which CNAME points to a given SmartConnect zone name.

During a disaster recovery scenario, CNAMEs complicate and extend the failover process, as many CNAMEs must be updated. Further, Active Directory Kerberos does not function with CNAMEs. Zone aliases are the recommended alternative.

OneFS provides an option for creating SmartConnect zone aliases. As a best practice, a SmartConnect zone alias should be created in place of CNAMEs. To create a SmartConnect zone alias, use the following command:

```
isi networks modify pool <subnet:pool> --add-zone-aliases=<zone alias name>
```

Once the SmartConnect zone alias is provisioned, a matching delegation record must be created in the site DNS, pointing to a SmartConnect Service IP (SSIP).

### 8.10.3 One name server record for each SmartConnect zone name or alias

One delegation for each SmartConnect zone name or each SmartConnect zone alias on a cluster is recommended. This method permits the failover of only a portion of the cluster's workflow—one SmartConnect zone—without affecting any other zones.

For example, an administrator may have the following delegations in place:

```
cls01-smb.foobar.com.      IN      NS      cls01-ssip.foobar.com
cls01-nfs.foobar.com.      IN      NS      cls01-ssip.foobar.com
cls01-hdfs.foobar.com.      IN      NS      cls01-ssip.foobar.com
```

With this approach, the administrator has the flexibility of failing over or moving one or multiple delegations.

As an example, consider the following:

```
cls01-smb.foobar.com.      IN      NS      cls01-ssip.foobar.com
cls01-nfs.foobar.com.      IN      NS      cls01-ssip.foobar.com
cls01-hdfs.foobar.com.     IN      NS      cls99-ssip.foobar.com
```

It is not recommended to create a single delegation for each cluster and then create the SmartConnect zones as sub-records of that delegation. As an example, consider the following:

```
smb.cls01.foobar.com
nfs.cls01.foobar.com
hdfs.cls01.foobar.com
```

The advantage of the process explained in this section is it enables PowerScale administrators to change, create, or modify their SmartConnect zones and zone names as needed without involving a DNS team. However, the disadvantage of this approach is that it causes failover operations to involve the entire cluster and affects the entire workflow, not just the impacted SmartConnect zone, \*.cls01.foobar.com.

#### 8.10.4 Multiple DNS resolvers in a groupnet

Complex data center integrations and authentication realms may require multiple DNS resolvers within a logical group. If possible, separate these into multiple groupnets to create a hierarchy aligning with the site environment.

Depending on the existing hierarchy, separating DNS instances in multiple groupnets may not be an option, requiring multiple resolvers and name servers to reside in a single groupnet. For these implementations, it is recommended to leverage DNS hosts that are capable of DNS forwarding to forward to the corresponding DNS resolvers.

OneFS allows up to three DNS instances in a single groupnet. Proceed with caution when adding more than a single DNS instance to a groupnet. Determining how clients are routed to a specific DNS instance impacts the client's performance and overall session latency.

The multiple DNSs in a single groupnet are managed in OneFS through the `isi network groupnets [create or modify]` command and the following options:

- `--dns-options`: The `dns-options` configuration specifies how OneFS selects between the multiple DNS instances. In the current release, the only option is to rotate between the instances, providing a round-robin DNS selection process.
- `--dns-search`: The `dns-search` configuration specifies up to three DNS suffixes in a single groupnet. OneFS searches these DNS suffixes to resolve queries.
- `--dns-servers`: The `dns-servers` configuration specifies up to three IP addresses for multiple DNS instances in a single groupnet.

For round-robin options of the DNS instances or searching through more than a single DNS suffix, it is crucial to consider the varying impacts on client experience. If the DNS for a specific session is not consistent, the overall session latency may increase depending on the assigned DNS. If the latency is inconsistent, workflow success is difficult to predict, resulting in users reporting random experiences. If a groupnet contains two DNS instances, this results in a low chance of having the appropriate DNS assigned

to a session. If three DNS instances are specified for a groupnet, the resolution success rate drops further.

## 8.11 SmartConnect in isolated network environments

SmartConnect is, effectively, a limited implementation of a custom DNS server: It answers only for the SmartConnect zone names or aliases configured on it. In order to use SmartConnect in an isolated network environment where no DNS infrastructure is available (such as a DMZ), configure the client systems using the SmartConnect service IP address as the primary DNS server. Configuring the client systems this way ensures the following:

- Requests to connect to PowerScale clusters with SmartConnect zone names will succeed
- The isolated network benefits from SmartConnect features, such as load-balancing and rerouting traffic to prevent unavailable nodes, will work as expected in a typical, non-isolated deployment.
- It is essential to recognize that PowerScale OneFS is not a full DNS server. Hence, it will only answer for SmartConnect Zones.
- Utilize SmartConnect Zones and aliases in place of name server zones. As an example, do not create name server zones on a bind host. Alternatively, create SmartConnect Zones and aliases.

The following commands show how to simulate and test a configuration that uses the SmartConnect service IP address as the primary DNS server.

```
C:\>nslookup
Default Server: 10.123.17.60
Address: 10.123.17.60

> isi01-s0.domain.com
Server: [10.123.17.60]
Address: 10.123.17.60

Name: isi01-s0.domain.com
Address: 10.123.17.64

> isi01-s0.domain.com
Server: [10.123.17.60]
Address: 10.123.17.60

Name: isi01-s0.domain.com
Address: 10.123.17.63
```

## 8.12 SmartConnect DNS, subnet, and pool design

This section provides a starting point for planning SmartConnect DNS, subnet, and SmartConnect pool layouts that meet the needs of most new cluster implementations with a SmartConnect Advanced License.

---

**Note:** SmartConnect Service IP Addresses (SSIPs) are only supported for use by a DNS server. Although SSIPs may be used in other configurations, the design intent was for a DNS server. Thus, other implementations with SSIPs are not supported.

---

### 8.12.1 SmartConnect zone naming

It is recommended to ensure SmartConnect Zones are named according to relevant details for clarity and simple recognition. For example, names should be composed of some or all of the following variables:

- **Cluster Name:** Active Directory (AD) uses the cluster name as the AD machine account name. For example, when a cluster named isi01 joins Active Directory, isi01 is the cluster's machine account name. Using the cluster/machine account name in all DNS entries simplifies cluster administration and troubleshooting.
- **IP Allocation Strategy:** Each SmartConnect zone has an IP allocation strategy set to static or dynamic. The allocation strategy is allocated in the zone name, for example, by using "d" for dynamic and or "s" for static.
- **SmartConnect Pool ID:** Each SmartConnect pool has a unique name or number that identifies it. By default, the first pool called on a cluster is pool0, the second is pool1, and so on. These identifiers are recommended to be part of the zone name.
- **SSIP:** Use the SSIP in the zone name to indicate a SmartConnect Service IP zone.

The variables above together form a SmartConnect zone name. For example: isi01-s0.domain.com

The name includes the cluster name (isi01), the allocation strategy of the zone ("s" for static), and the number of the pool (pool0).

For example, a cluster with three pools:

- pool0: Static for client I/O for stateful protocols
- pool1: Dynamic for client I/O for stateless protocols
- pool2: Static for Backup and Replication

Based on the SmartConnect zone, pool, and the cluster information, the following is a sample DNS layout for the cluster named 'isi01':

- isi01-ssip.domain.com in [A] to 10.x.y.z
- isi01-s0.domain.com in [NS] to isi01-ssip.domain.com
- isi01-d1.domain.com in [NS] to isi01-ssip.domain.com
- isi01-s2.domain.com in [NS] to isi01-ssip.domain.com

### 8.12.2 SmartConnect with multiple node pools or types

From a client or application perspective, the goals for all scale-out NAS deployments are consistency and availability. Consistency, in this context, implies that every time a client connects, whether that client is an application server or a user opening their home directory, the same level of performance is provided. Dell EMC offers a number of different PowerScale node types with varied performance profiles.

Many factors determine performance in network-attached storage. In a PowerScale cluster, key components are the front-end performance, which consists of the network card, CPU, and memory in the node that is serving the relevant data protocol, and the back-end performance, which, in this case, is the disk tier or pool where the data resides. In the context of SmartConnect configuration, creating a connection pool that spans across different node performance levels is not recommended. For example, a pool with Isilon F800 nodes and A200 nodes would provide significantly varying protocol performance. It is imperative to understand how the nodes within a connection pool impact client experience.

## 8.13 Where the SmartConnect Service IP (SSIP) runs (pre OneFS 8.2)

The SSIP service is updated for OneFS 8.2; this section is specific to releases prior to OneFS 8.2. For more information on the SSIP in OneFS 8.2, refer to Section 8.14, SmartConnect Multi-SSIP..

The PowerScale clustered compute, and storage platform has no single point of failure. However, the SmartConnect DNS service must be active on only one node at any time, per subnet. The SmartConnect Service IP resides on the node with the lowest node ID that has an interface in the given subnet, not necessarily on the node with the lowest Logical Node Number (LNN) in the cluster.

To illustrate how this works, suppose that an existing four-node cluster is refreshed with four new nodes. Assume that the cluster has only one configured subnet, all the nodes are on the network, and that there are sufficient IP addresses to handle the refresh. The first step in the cluster refresh is to add the new nodes with the existing nodes, temporarily creating an eight-node cluster. Next, the original four nodes are SmartFailed. The cluster is then composed of the four new nodes with the original data set.

As the administrators perform the refresh, they check the current configuration using the **isi config** command, with the status advanced command, as shown in the following example:

```
isi config
>status advanced
```

The SmartConnect service continues to run throughout the process as the existing nodes are refreshed. The following example illustrates where the SmartConnect service runs at each step in the refresh process.

Once the four new nodes are added to the cluster, based on the existing naming convention, they are automatically named clustername-5, clustername-6, clustername-7, and clustername-8. At this point, the Node IDs and LNNs are displayed in the following table:

Table 5 8-node cluster configuration, before SmartFail

Logical Node Number (LNN)	NodeID	Node Name	New or Original Node
1	1	clustername-1	Original
2	2	clustername-2	Original
3	3	clustername-3	Original
4	4	clustername-4	Original
5	5	clustername-5	New
6	6	clustername-6	New
7	7	clustername-7	New
8	8	clustername-8	New

**Note:** The SmartConnect service always runs on the node with the lowest node ID; at this point, NodeID 1 is mapping to LNN 1.

Next, the original nodes are removed using SmartFail. The updated Node IDs and LNNs are displayed in the following table:

Table 6 4-node cluster configuration, after SmartFail

Logical Node Number (LNN)	Node ID	Node Name	New or Original Node
1	5	clustername-5	New
2	6	clustername-6	New
3	7	clustername-7	New
4	8	clustername-8	New

---

**Note:** The SmartConnect service always runs on the node with the lowest node ID; at this point, NodeID 5 is mapping to LNN 1.

---

Keeping the naming convention consistent, the administrators re-name the new nodes, formerly clustername-5, clustername-6, clustername-7, and clustername-8, to clustername-1, clustername-2, clustername-3, and clustername-4, respectively. The updated Node IDs and LNNs remain the same, but map to a different Node Name, as displayed in the following table:

Table 7 4-node cluster configuration – after re-name

Logical Node Number (LNN)	Node ID	Node Name	New or Original Node
1	5	clustername-1	New
2	6	clustername-2	New
3	7	clustername-3	New
4	8	clustername-4	New

---

**Note:** The SmartConnect service always runs on the node with the lowest node ID; at this point, NodeID 5 is mapping to LNN 1.

---

If LNN 1 is offline for maintenance, the SmartConnect service migrates to LNN 2, because LNN 2 has the next lowest NodeID number, 6.

## 8.14 SmartConnect Multi-SSIP

PowerScale OneFS 8.2 introduces support for more than one SSIP per subnet. In previous releases, only a single SSIP per subnet was supported and resided on the lowest available NodeID, as explained in Section 8.13, Where the SmartConnect Service IP (SSIP) runs (pre OneFS 8.2).

. The dependence on a single SSIP caused problems during node maintenance, reboots, or interface flaps. The complications are further magnified, considering the lowest available NodeID is usually the node that is rebooted or is scheduled for maintenance first.

The addition of more than a single SSIP provides fault tolerance and a failover mechanism, ensuring the SmartConnect service continues to load balance clients according to the selected policy. In previous releases of OneFS, once the node hosting the SSIP was out of service, or if the interface was flapping, client connections would fail momentarily as the SSIP migrated to a different node.

The number of SSIPs available per subnet depends on the SmartConnect license. SmartConnect Basic allows 2 SSIPs per subnet while SmartConnect Advanced allows 6 SSIPs per subnet, as displayed in Table 8.

Table 8 SmartConnect SSIPs by license

SmartConnect License	Basic	Advanced
SSIPs per subnet	2	6

It is important to recognize that SmartConnect Multi-SSIP is not an additional layer of load balancing for client connections. Additional SSIPs provide redundancy and reduce failure points in the client connection sequence. Reverting to the original figure explaining SmartConnect's connection sequence, additional connections are added at step 2, as illustrated in Figure 17.

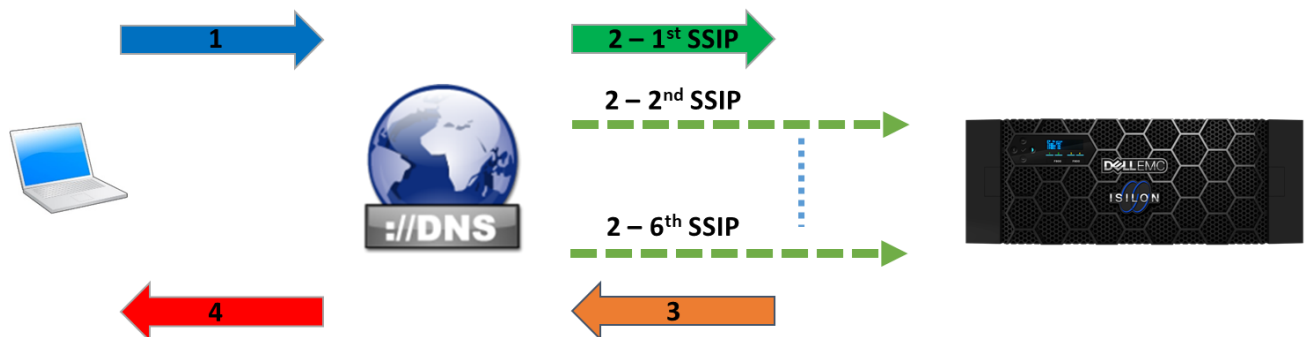


Figure 17 SmartConnect Multi-SSIP sequence

At step 2, the site DNS server sends a DNS request to the SSIP and awaits a response in step 3 for a node's IP address based on the client connection policy. If for any reason, the response in step 3 is not received within the timeout window, the connection times out. The DNS server tries the 2nd SSIP and awaits a response in step 3. After another timeout window, the DNS server continues cycling through subsequent SSIPs, up to the sixth SSIP with SmartConnect Advanced, if a response is not received after a request is sent to each SSIP.

Although the additional SSIPs are in place for failover, the SSIPs configured are active and respond to DNS server requests. The Multi-SSIP configuration is Active-Passive, where each node hosting an SSIP is



independent and ready to respond to DNS server requests, irrespective of the previous SSIP failing. Therefore, SmartConnect continues to function correctly if the DNS server contacted the other SSIPs, providing SSIP fault tolerance. However, as each node hosts an SSIP independent of the other SSIP hosting nodes, it is unaware of the current status of the load balancing policy and starts the load balancing policy back to the first option. For example, if the SmartConnect load balancing policy is round-robin for a 50-node subnet, assume the 1st SSIP has distributed IP addresses for the first ten nodes. If the 2nd SSIP is contacted by the DNS server, it starts distributing node IP addresses at the first option again, in this case, node one, rather than node eleven. The node hosting the SSIP is unaware of the node IP address distributed by the previous SSIP.

---

**Note:** As a best practice, do not configure the site DNS server to load balance the SSIPs. Each additional SSIP is only a failover mechanism, providing fault tolerance and SSIP failover. Allow OneFS to perform load balancing through the selected SmartConnect policy, ensuring effective load balancing.

---

### 8.14.1 Configuring OneFS for SmartConnect Multi-SSIP

Multi-SSIP is configured from the user interface or the CLI, by specifying a range of IP addresses. The range of IP addresses is applied to between 2 and 6 SSIPs per subnet, depending on the SmartConnect license.

To configure Multi-SSIP from the user interface, click **Cluster Management > Network Configuration**. Next, either select an existing subnet and click **Edit**, or if under a groupnet, click **More > Add subnet** and scroll to the **SmartConnect service IPs** section, as displayed in Figure 18.

Figure 18 SmartConnect Multi-SSIP user interface configuration

To configure Multi-SSIP from the CLI, use the `--sc-service-addr` option with an IP address range, as displayed in the following command: `isi network subnets modify subnet0 --sc-service-addr=192.168.25.10-192.168.25.11`

Additionally, the IP address range may be cleared, or additional ranges may be added, using the following commands:

```
isi network subnets modify subnet --help | grep sc-service-addr
[--sc-service-addr <ip_address_range> | --clear-sc-service-addr |
  --add-sc-service-addr <ip_address_range> | --remove-sc-service-addr
--sc-service-addr <ip_address_range>...
  requests. Specify --sc-service-addr for each additional IP address.
--clear-sc-service-addr
--add-sc-service-addr <ip_address_range>...
  Add items to list of IP addresses. Specify --add-sc-service-addr for
--remove-sc-service-addr <ip_address_range>...
  --remove-sc-service-addr for each additional IP address to remove.
```

## 8.14.2 Configuring a DNS server for SmartConnect multi-SSIP

Multi-SSIP is a feature for SSIP failover, providing SSIP fault tolerance. It is important to configure DNS servers for SSIP failover, ensuring the next SSIP is only contacted if the first SSIP connection times out. If the SSIPs are not configured in a failover sequence, the SSIP load balancing policy resets each time a new SSIP is contacted. The SSIPs do not track the current distribution status of the other SSIPs, as they function independently, negating the function of the selected load balancing policy.

Configuring IP addresses as a failover only addresses is not supported on all DNS servers. To support Multi-SSIP as a failover only option, a DNS server with support for failover addresses is recommended. If a DNS server does not support failover addresses, Multi-SSIP still provides advantages over a single SSIP. However, increasing the number of SSIPs may impact SmartConnect's ability to load balance.

---

**Note:** If the DNS server does not support failover addresses, test Multi-SSIP in a lab environment mimicking the production environment to confirm the impact on SmartConnect's load balancing for a specific workflow. Only after confirming workflow impacts in a lab environment should a production cluster be updated.

---

### 8.14.2.1 DNS servers supporting failover IP addresses

If the site DNS server supports failover IP addresses, proceed with the configuration in this section.

As explained earlier in Section 8.10, DNS delegation best practices, the first SSIP should be created in DNS as an address (A) record, also referred to as a host entry. The additional SSIPs should be configured as DNS A record failover IP addresses. The first IP address should point to the first SSIP, followed by each configured SSIP IP addresses for failover. The additional SSIPs provide redundancy in an Active-Passive pattern.

### 8.14.2.2 DNS servers without failover IP address support

If the site DNS server does not support failover IP addresses, proceed with the configuration in this section.

---

**Note:** Prior to configuring a DNS server that does not support failover IP addresses, consider the load balancing status in SmartConnect is independently managed by each SSIP, as explained in Section 8.14, SmartConnect Multi-SSIP. The total impact on load balancing behavior is dependent on if the site DNS server has recursion enabled, how many SSIPs are configured, the load balancing policy, and the workflow. To confirm the impacts in a specific environment test in a lab environment mimicking the production environment, prior to updating a production cluster.

---

To configure a DNS server for Multi-SSIP that does not support failover IP addresses, create an NS record, and matching A/AAAA record for each SSIP. Most DNS servers use a Round Trip Time (RTT) to decide which nameserver to utilize. As an example, for OneFS and a BIND DNS server, consider the following configuration:

OneFS configuration:

```
isi network subnets modify groupnet0.subnet0 --sc-service-name=cluster-
ns1.company.com --sc-service-addr=1.2.3.4-1.2.3.6
isi network pools modify groupnet0.subnet0.pool0 --sc-connect-policy round_robin
--sc-dns-zone cluster.company.com
```

BIND configuration:

```
cluster-ns1 IN A 1.2.3.4
cluster-ns2 IN A 1.2.3.5
cluster-ns3 IN A 1.2.3.6
```

```
$ORIGIN cluster.company.com.
@ IN NS cluster-ns1.company.com.
@ IN NS cluster-ns2.company.com.
@ IN NS cluster-ns3.company.com.
```

The configuration above may be adapted to Windows DNS servers or other DNS servers. The issue with Windows DNS server is the forced 1 second TTL, which impacts single SSIP configurations also, as noted in Section 8.16, Other SmartConnect considerations.

Additionally, in an environment where the site DNS server does not support failover IP addresses, consider the following:

- If the site DNS server has recursion enabled, consider that the nameservers may be contacted in a round-robin fashion. To confirm this behavior, check for a frequently changing nameserver through logging in OneFS. If the SmartConnect zone is configured for round-robin, try repeatedly querying the zone. If the DNS server returns an IP the same number of times as SSIPs configured, it is contacting nameservers in a round-robin configuration.
- If a site DNS server is not very sticky in terms of how it chooses name servers, load balancing will decrease as the number of SSIPs in a subnet increase. For example, consider the difference between the site DNS server returning the same IP two times in a row when two SSIPs are configured, and the site DNS server returning the same IP six times in a row when six SSIPs are configured.
- The selected SmartConnect load balancing policy is not round-robin, having multiple SSIPs and a site DNS server that is not sticky in deciding what SSIP to use can exacerbate the load balancing problem. This could result in more clients than expected, landing on the lightest weighted node.
- If the workload consists of high throughput, usage, or demanding clients, using Multi-SSIP makes the above considerations significantly more noticeable. On the contrary, if the workload consists of many smaller client connections, the impact of Multi-SSIP on Load Balancing may go unnoticed especially with round-robin policies given that SmartConnect eventually distributes each node's IP address almost an equal number of times.

### 8.14.3 SSIP node assignment

Within a subnet, up to six SSIPs are available, depending on the SmartConnect license. Prior to OneFS 8.2, the single SSIP was assigned to the lowest Node ID in the specified subnet. Hosting the SSIP on the lowest Node ID created issues as in many cases, the lowest Node ID is providing other services and could be the first to reboot in a rolling upgrade.

Multi-SSIP introduces an enhancement to assigning SSIPs. Attaching an SSIP to a node is no longer dependent on the Node ID. OneFS 8.2 creates a file containing SSIP information, the SSIP Resource File. In order to host an SSIP, a node must hold a lock on this file. All the nodes that are ready to host an SSIP, attempt to lock the SSIP Resource File. The first nodes to get the lock, host the SSIP. The new process ensures the node assignment is based on a lock to nodes within the subnet, avoiding the issues from previous releases. Once the node is offline, or the interface goes down, the SSIP becomes available for lock again and the next quickest node to capture the lock hosts the SSIP, as illustrated in Figure 19. OneFS ensures that SSIPs are as evenly distributed as possible within a subnet, utilizing a feature to limit a single node from hosting multiple SSIPs. In certain scenarios, a node may host more than a single SSIP, depending on the number of nodes and SSIPs in the subnet.



Figure 19 SSIP Resource File

OneFS 8.2 also introduces a new method for handling configuration and group changes. In releases prior to OneFS 8.2, SmartConnect unconditionally stopped and unconfigured the SSIP during a configuration or group change, and then evaluated where it should run, which was frequently the same node. In OneFS 8.2, the SSIP remains in place through configuration and group changes. After the changes, the SSIP moves only if necessary, minimizing failover impacts.

To confirm which of the nodes are hosting SSIPs, use the following commands:

```
isi_for_array ifconfig | grep <SSIP>
isi_for_array ifconfig | grep "zone 0"
```

## 8.15 DNS and time-to-live

The OneFS SmartConnect DNS server is designed to respond to delegated queries from a site DNS server for SmartConnect zones defined on the cluster. For load-balancing to be effective, it is critically important that the site DNS servers do not cache the results.

On the cluster side, the SmartConnect time-to-live (TTL) is configurable. The default is zero, and should not be changed for normal use cases. To configure the TTL for a SmartConnect pool, use the following command:

```
isi network pools modify <pool id> --sc-ttl=0
```

For load-balancing to operate optimally, it is also important that the site DNS honors the TTL returned by the OneFS SmartConnect DNS server. If the site DNS “clamps the TTL”, it is possible for the site DNS server to erroneously return the same cached value if multiple client requests are received within the same “clamp window”.

### 8.15.1 Microsoft Windows DNS

Windows Server DNS 2003, 2008, 2012, 2016, and 2019, clamp the minimum TTL to one second. If many client requests are expected within a one-second timeframe, consider a different DNS server.

### 8.15.2 BIND DNS

Newer versions of BIND have introduced features that impact SmartConnect’s ability to load balance. Depending on the BIND DNS version, consider the following:

- BIND 9.12 introduced serve-stale functionality, which allows DNS resource records with an expired TTL to be returned if the DNS server is failing to resolve. Under normal operation, this feature does not cache records. However, during a SmartConnect error, if DNS records are still in the cache, this may impact load balancing.
- BIND 9.10.3/9.11 introduced new options to prevent DDOS attacks. The options include “max-clients-per-query” and “clients-per-query,” which bundle identical queries from clients and only send a single query to a recursive name server. The “clients-per-query” option is the baseline for when to start bundling, and “max-clients-per-query” is the cap on how many clients to bundle. For more details on these options, refer to <https://www.isc.org/blogs/tldr-resolver-ddos-mitigation/>.
- BIND 9.11 added options for “fetches-per-server” and “fetches-per-zone”. “Fetches-per-server” is a self-tuning option that limits the number of outgoing requests to individual name servers. “Fetches-per-zone” limits the number of outstanding requests per zone. For more details on BIND 9.11, refer to <https://bind.isc.org/doc/arm/9.11/Bv9ARM.html>.

---

**Note:** In an event where these limits are engaged, multiple clients receive the same IP address, negating SmartConnect’s ability to load balance connections. As a best practice, test these limits in a lab environment and understand how each of these features impacts a specific workflow.

---

## 8.16 Other SmartConnect considerations

During SmartConnect configuration, consider the following points:

- It is recommended to disable client DNS caching, when possible. To handle client requests properly, SmartConnect requires that clients use the latest DNS entries. If clients cache SmartConnect DNS information, they could connect to incorrect SmartConnect zone names. In this event, SmartConnect might not appear to be functioning correctly.
- If traffic is traversing firewalls, ensure that the appropriate ports are open. For example, if UDP port 53 is opened, also ensure TCP port 53 is opened.
- Certain clients perform DNS caching and might not connect to the node with the lowest load if they make multiple connections within the lifetime of the cached address. For example, this issue occurs on macOS X for certain client configurations.
- The site DNS servers must be able to communicate with the node that is currently hosting the SmartConnect service.
- Site DNS servers might not exist in the regular local subnets, or in any of the subnets that clients occupy. To enable the SmartConnect lookup process, ensure that the DNS servers use a consistent route to the cluster and back. If the site DNS server sends a lookup request that arrives through one local subnet on the cluster, but the configured cluster routing causes the response to be sent through a different subnet, it’s likely that the packet will be dropped, and the lookup will fail. The solutions and considerations for SmartConnect are similar to the client scenarios. Additionally, the DNS server might benefit from a static route to the subnet that contains the SSIP address or addresses.
- SmartConnect makes it possible for different nodes to have different default routes, but this is fundamentally determined by connectivity. SmartConnect enables administrators to define multiple gateways, with 1 gateway per subnet. Each gateway is assigned a priority when it is defined. On any node, SmartConnect attempts to use the highest priority gateway—the gateway that has the lowest number—that has an available functioning interface in a subnet that contains the gateway address.

# 9 Ethernet, MTU, and IP overhead

A Maximum Transmission Unit (MTU) is the largest packet size or frame, which may be sent along a link. The MTU is specified in octets and is utilized by TCP to determine the maximum size of a packet per transmission. A large MTU provides less overhead as packet headers and acknowledgments are not consuming space and bandwidth. However, this could lead to retransmissions or drops if a hop does not support it. On the contrary, a small MTU is not as efficient as overhead increases with packet headers and acknowledgments.

Generally speaking, the MTU across the internet is 1500 bytes. As such, most devices limit packet size to roughly 1472 bytes, allowing for additional overhead and remaining under the 1500-byte limit. Additional overhead may be added as the packet goes through different hops. The IEEE 802.3 standard also specifies 1500 bytes as the standard payload.

## 9.1 Ethernet packet

An Ethernet frame carries a payload of data and is carried by an Ethernet packet. The frame could be IPv4 or IPv6 and TCP or UDP. The IEEE 802.3 standard defines the structure of each packet. As a packet traverses different layers, the structure is modified accordingly. In the diagram below, the structure is displayed as it would traverse the wire, or Layer 1. Dissecting how a packet is structured on the wire lends to an understanding of how the packet overhead is impacted and all of the other components required to send a payload.

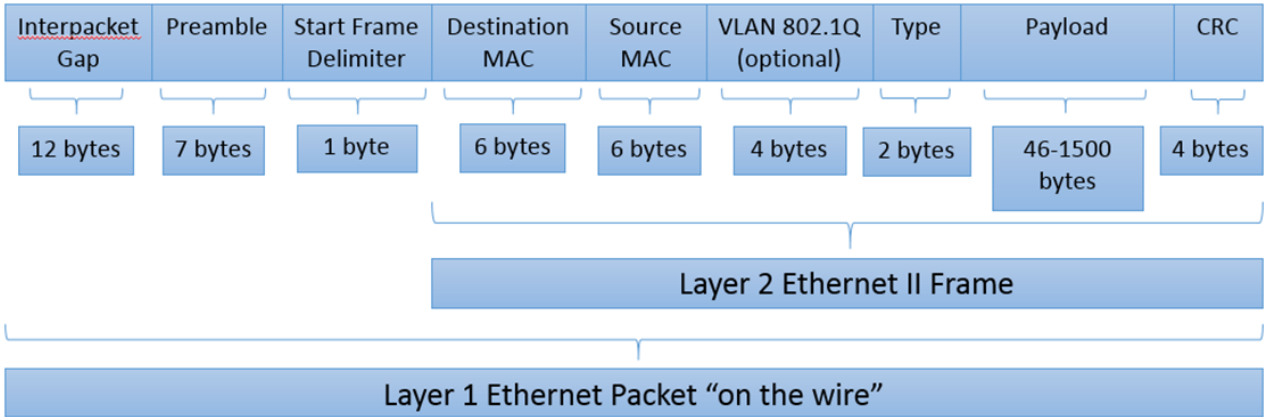


Figure 20 Ethernet packet

An Ethernet packet on the wire at Layer 1 is composed of the following fields:

- Interpacket Gap: Serves as a gap between each frame, similar to a spacer. The Interpacket gap is only part of Layer 1. The field originates from a time when hubs were common, and collisions were more commonplace.
- Preamble: Composed of alternating 1 and 0 bits for receiver clock synchronization. The Preamble is only part of Layer 1.
- Start Frame Delimiter: Identifies the start of an Ethernet frame.
- Destination MAC: Contains the MAC address of the destination station for which the data is intended.
- Source MAC: Contains the MAC address of the sending station.
- VLAN 802.1Q: Optional field utilized if a VLAN is identified.
- Type: Also known as the EtherType field, this defines the type of protocol that is encapsulated in the payload. In the example above, it is an Ethernet II Frame, the most widely accepted type.



- **Payload:** Spans from 46 to 1500 bytes and contains user data. If it is smaller than 46 bytes, blank values are entered to bring this up to 46 bytes as it is the minimum value. The Payload consists of protocol data for TCP, UDP or RTP and IPv4 or IPv6. The next section explains the Payload field in greater depth.
- **CRC:** Cyclic Redundancy Check is part of the Frame Check Sequence (FCS) to detect errors within the frame. The CRC code should result in a zero if the data does not contain any errors.

## 9.2 Ethernet payload

The Ethernet payload varies based on the type of data it is carrying. It is a combination of either TCP, UDP, or RTP header combined with an IPv4 or IPv6 header, and most importantly the actual payload which contains the data that is being sent. The fields within the payload are displayed in Figure 21.

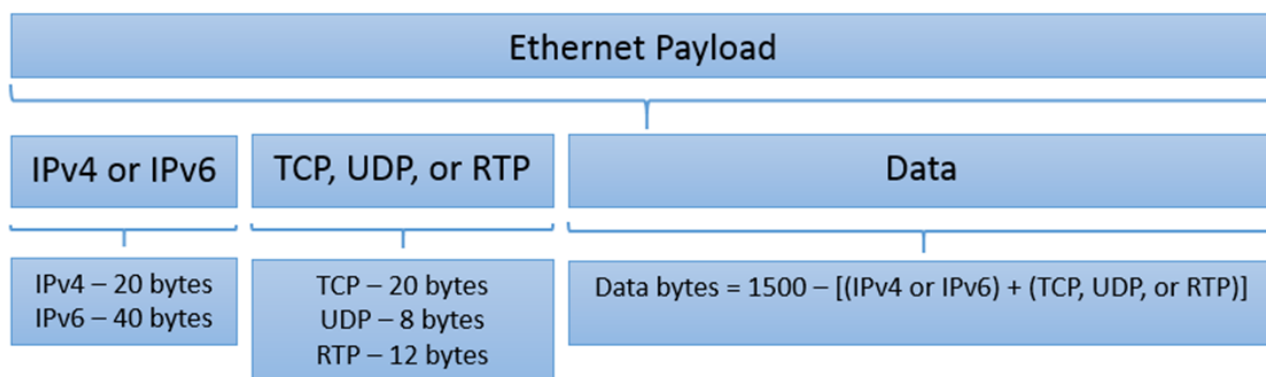


Figure 21 Ethernet payload options

As displayed in Figure 21, the amount of actual data sent within an Ethernet Frame is dependent upon the number of bytes consumed by the other fields. Other options are available which are not listed here. For example, Linux hosts automatically add a timestamp to the TCP stack, adding 12 bytes.

## 9.3 Jumbo frames

Jumbo frames are Ethernet frames where the MTU is greater than the standard 1500 bytes and a maximum of 9000 bytes. The larger MTU provides greater efficiency as less overhead and fewer acknowledgments are sent across devices, drastically reducing interrupt load on endpoints. Jumbo frames are recommended for most workloads as the amount of data sent per message is far greater, reducing processing times and maximizing efficiency. While the general assumption is that Jumbo frames provide performance advantages for all workloads, it is important to measure results in a lab environment simulating a specific workload to ensure performance enhancements.

For Jumbo frames to take advantage of the greater efficiencies, they must be enabled end-to-end on all hops between endpoints. Otherwise, the MTU could be lowered through PMTUD or packets could be fragmented. The fragmentation and reassembly impact the CPU performance of each hop, which impacts the overall latency.

For example, if a client is set to an MTU of 1500 bytes while other hops are set to 9000 bytes, transmission along the path will most likely be set to 1500 bytes using PMTUD, unless other options are configured.

Jumbo frames utilize the same Ethernet packet structure described in the previous section. However, the difference is the size of the data within the payload. As the byte consumption of the other components within

the frame remain the same, each packet contains more data with the same overhead. A Jumbo frame Ethernet payload is displayed in the following figure:

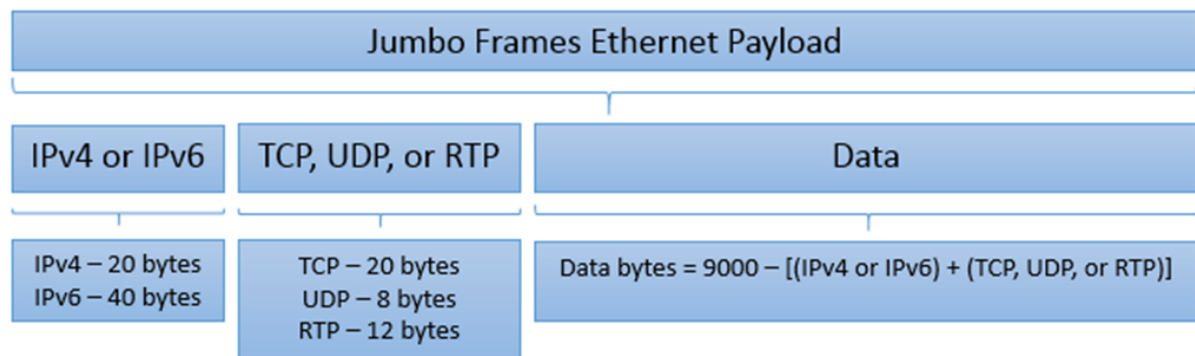


Figure 22 Jumbo frames Ethernet payload

## 9.4 IP packet overhead

PowerScale nodes utilize 10, 25, and 40 GbE NICs for front-end networking. In order to maximize throughput on these high bandwidth links, Jumbo frames are recommended for greater throughput. Standard 1500 byte and Jumbo 9000-byte packets are formed with the same packet structure at Layer 1, with the only difference pertaining to the actual data payload. Although the overhead is identical for standard and Jumbo packets, the ratio of the data to the overhead varies significantly.

For every payload sent to Layer 1 on the wire, the following fields are required:

Interpacket Gap / Preamble / Start Frame Delimiter / Destination MAC / Source MAC / Type / CRC

In bytes, this translates to:

$$12 + 7 + 1 + 6 + 6 + 2 + 4 = 38 \text{ bytes}$$

Hence, regardless of the payload fields, every payload requires an additional 38 bytes to be sent. It is important to note that this does not consider the optional VLAN tag, which requires an additional 4 bytes. The following sections provide examples of packet overhead based on the payload fields.

### 9.4.1 Example 1: Standard 1500-byte payload – IPv4/TCP

IPv4 and TCP headers consume the following bytes:

$$20 \text{ bytes (IPv4)} + 20 \text{ bytes (TCP)} = 40 \text{ bytes}$$

If the payload headers consume 40 bytes, the data field for a standard 1500-byte payload consumes:

$$1500 - 40 = 1460 \text{ bytes}$$

Therefore, a standard 1500-byte packet with IPv4 and TCP headers results in a data to Ethernet frame percentage as follows:

$$\frac{(\text{Data Bytes})}{(\text{Total Ethernet Frame Bytes})} = \frac{(1500 - 40)}{(1500 + 38)} = \frac{1460}{1538} = .949 \Rightarrow 94.9\%$$



### 9.4.2 Example 2: Jumbo 9000-byte payload – IPv4/TCP

A standard 9000-byte payload that contains IPv4 and TCP headers consumes the following bytes:

20 bytes (IPv4) + 20 bytes (TCP) = 40 bytes

If the payload headers consume 40 bytes, the data can field consumes:

9000 - 40 = 8960 bytes

Therefore, a standard 1500-byte packet with IPv4 and TCP headers results in a data to Ethernet frame percentage as follows:

$$\text{Data Bytes} / (\text{Total Ethernet Frame Bytes}) = (9000 - 40) / (9000 + 38) = 8960/9038 = .991 \Rightarrow 99.1\%$$

### 9.4.3 Example 3: Standard 1500-byte payload – IPv4/TCP/Linux timestamp

Linux automatically inserts the timestamp within the payload. A standard 1500-byte payload that contains IPv4, TCP, and timestamp headers consumes the following bytes:

20 bytes (IPv4) + 20 bytes (TCP) + 12 bytes (timestamp) = 52 bytes

If the payload headers consume 52 bytes, the data field consumes:

1500 - 52 = 1448 bytes

Therefore, a standard 1500-byte packet with IPv4 and TCP headers results in a data to Ethernet frame percentage as follows:

$$(\text{Data Bytes}) / (\text{Total Ethernet Frame Bytes}) = (1500 - 52) / (1500 + 38) = 1448/1538 = .941 \Rightarrow 94.1\%$$

### 9.4.4 Example 4: Jumbo 9000-byte payload – IPv4/TCP/Linux timestamp

Linux automatically inserts the timestamp within the payload. A standard 9000-byte payload that contains IPv4, TCP and timestamp headers consumes the following bytes:

20 bytes (IPv4) + 20 bytes (TCP) + 12 bytes (timestamp) = 52 bytes

If the payload headers consume 52 bytes, the data field consumes:

9000 - 52 = 8948 bytes

Therefore, a standard 1500-byte packet with IPv4 and TCP headers results in a data to Ethernet frame percentage as follows:

$$(\text{Data Bytes}) / (\text{Total Ethernet Frame Bytes}) = (9000 - 52) / (9000 + 38) = 8948/9038 = .990 \Rightarrow 99.0\%$$

## 9.5 Data payload to Ethernet frame efficiency

Utilizing the calculations above, the table below provides additional examples of the amount of data that is sent per Ethernet frame for standard and Jumbo frames.

Table 9 Data payload to Ethernet frame percentage

Packet Type	Data to Ethernet Frame Percentage	
	Standard Frame	Jumbo Frame
IPv4 / TCP	94.93%	99.14%
IPv4 / TCP / Linux Timestamp	94.15%	99.00%
IPv4 / TCP / Linux Timestamp / VLAN	93.90%	98.96%
IPv6 / TCP	93.63%	98.92%
IPv6 / TCP / Linux Timestamp	92.85%	98.78%
IPv6 / TCP / Linux Timestamp / VLAN	92.59%	98.74%
IPv4 / UDP	95.71%	99.27%
IPv4 / UDP / Linux Timestamp	94.93%	99.14%
IPv4 / UDP / Linux Timestamp / VLAN	94.67%	99.09%
IPv6 / UDP	94.41%	99.05%
IPv6 / UDP / Linux Timestamp	93.63%	98.92%
IPv6 / UDP / Linux Timestamp / VLAN	93.37%	98.87%

**Note:** NFS v2 is UDP. NFS v3 and v4 are TCP. SMB is TCP.

As displayed in the table above, Jumbo frames deliver between 98%-99% efficiency depending on the packet type. The efficiencies are only maximized when all hops from the client endpoint to a PowerScale node support Jumbo frames. Otherwise, packets may be fragmented leading to additional processing overhead on devices or PMTUD finding the lowest MTU along the path. Therefore, Jumbo frames are recommended for optimal performance. However, it is important to recognize that each workload environment is unique and measuring performance enhancements in a lab are recommended prior to a production network update.

## 9.6 ICMP and MTU with OneFS

Network devices employ Internet Control Message Protocol (ICMP) to gather communications related information. ICMP is capable of sending error messages but also delivers operational information. Ping and TraceRoute both send ICMP messages to provide connectivity information including latency and network hops.

Most devices have a default MTU that is configurable and remains at the defined value. PowerScale OneFS determines the MTU specific to each transaction. After the initial TCP handshake, the PowerScale node

sends an ICMP message for Path MTU Discovery (PMTUD), RFC 1191, gathering the maximum supported MTU. If for any reason ICMP is disabled, or PMTUD is not supported, this causes OneFS to default the MTU to 536 bytes, which typically leads to performance degradation.

## 9.7 OneFS MTU commands

To check the current configured MTU, enter the following command:

```
isi networks subnets list -v
```

To modify the MTU, use the isi command with the following context:

```
isi network subnets modify groupnet0.subnet1 --mtu=1500 --gateway=198.162.100.10
--gateway-priority=1
```

## 9.8 VLAN and interface MTU

VLAN specific MTUs ensure a consistent MTU across all network device hops in a session. OneFS allows multiple VLANs on a single interface, providing support for multiple workloads. The interfaces support multiple VLANs and the associated MTUs for those VLANs. A VLAN's MTU must be less than or equal to the parent interface's MTU. If the parent interface isn't explicitly configured, it will inherit the MTU of the VLAN with the greatest MTU.

## 9.9 Confirming transmitted MTU

Manually checking a permitted MTU ensures a configured MTU is transmitted. The ping command is used to confirm if an MTU can be transmitted. It is recommended to start with the largest MTU and work down to find the limit.

For example, to check if an MTU of 8900 bytes is transmitted to an endpoint, from the OneFS CLI, use the following command: `ping -s 8900 -D <IP Address>`. The '-s' specifies the packet size, and the '-D' specifies the not to fragment the packet.

If the ping is successful, the MTU is transmitted across. If the ping is unsuccessful, gradually lower the MTU until it is successfully transmitted. Confirm the MTU can be transmitted from both endpoints.

OneFS is based on FreeBSD. FreeBSD also has options for gradually increasing the MTU by performing a 'sweeping ping' using the `-g` option. For more information on ping options in FreeBSD, access the FreeBSD manual at the following link: [https://www.freebsd.org/cgi/man.cgi?ping\(8\)](https://www.freebsd.org/cgi/man.cgi?ping(8))

## 10 Access Zones best practices

When Access Zones are configured, a root-based path must be defined to segment data into the appropriate Access Zone and enable the data to be compartmentalized. Access Zones carve out access to a PowerScale cluster creating boundaries for multi-tenancy or multi-protocol. They permit or deny access to areas of the cluster. At the Access Zone level, authentication providers are also provisioned.

### 10.1 System Zone

When a PowerScale cluster is first configured, the System Zone is created by default. The System Zone should only be used for management as a best practice. In certain special cases, some protocols require the system zone, but generally speaking, all protocol traffic should be moved to an Access Zone. If nothing else, NFS and SMB should have protocol specific Access Zones.

Moving client traffic to Access Zones ensures the System Zone is only used for management and accessed by administrators. Access Zones provide greater security as administration, and file access is limited to a subset of the cluster, rather than the entire cluster.

### 10.2 Root Based Path

SmartConnect zones map to Access Zones, which map to a Root Based Path. When an Access Zone is defined, a Root Based Path must be defined. Best practice is to use the cluster name, a numerical Access Zone number, and a directory. For example, Access Zone 1 maps to `/ifs/clustername/az1/<data directories>`, Access Zone 2 maps to `/ifs/clustername/az2/<data directories>`. A Root Based Path with this delineation, provides data separation, Multi-Tenancy, maintains the Unified Permission model and makes SyncIQ failover and fallbacks easier.

Generally, the best practice is to remove all data access from the default System Zone. Otherwise, this leads to complications in the future as the cluster grows and additional teams or workflows are added. Further, as mentioned above, create a subdirectory under the Access Zone, rather than using the root of the zone, as this makes migration and disaster recovery simpler. It is preferred not to have an overlap of Root Based Paths unless it is required for a specific workflow. Overlap is supported in 8.0 and newer releases through the CLI.

In the figure below, as Cluster 1 fails over to Cluster 2, the directory structure remains consistent, easily identifying where the files originated from. This delineation also ensures clients have the same directory structure after a failover. Once the IP address is updated in DNS, the failover is transparent to clients. As more clusters are brought together with SyncIQ, this makes it easier to manage data, understanding where it originated from and provides seamless disaster recovery.

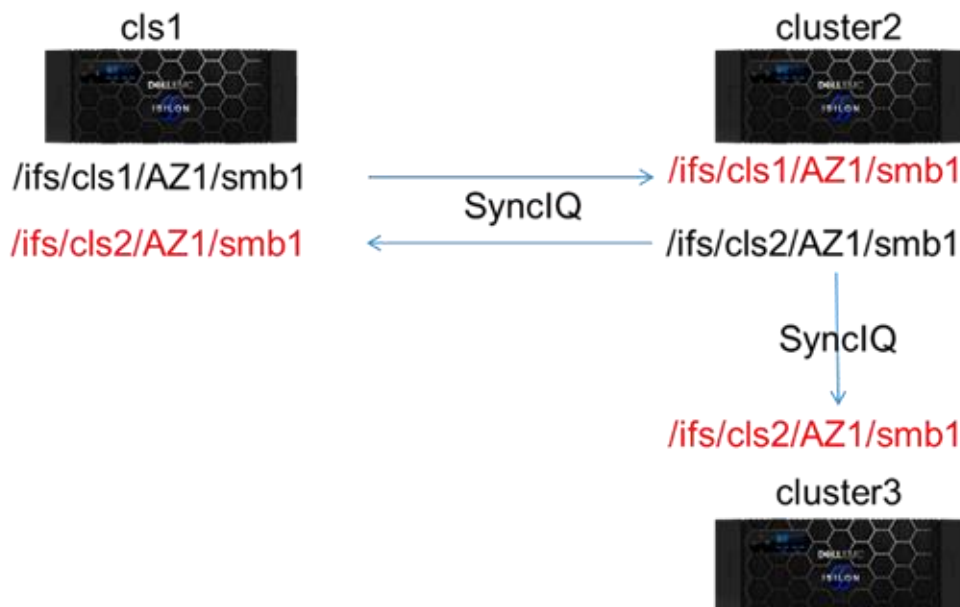


Figure 23 Importance of Root Based Path best practices

Root Based Paths may also be based on protocol. As an example, protocols are matched with a Root Based Path in the following table:

Table 10 Protocol-specific Access Zones

Protocol	Root Based Path
NFS Access	<code>/ifs/cls1/AZ1/nfs</code>
SMB Access	<code>/ifs/cls1/AZ2/smb</code>
NFS / SMB / HDFS	<code>/ifs/cls1/AZ3/mp</code>

# 11 Source-Based Routing considerations

Source-Based Routing (SBR) with PowerScale OneFS is discussed in the [PowerScale OneFS 8.1.0 External Network Connectivity Guide](#). This section clarifies how SBR functions. The naming convention suggests that SBR is routing packets based on a source IP address. However, SBR is actually a mechanism to dynamically create per-subnet default routes. The router used as this gateway is derived from the subnet configuration. Gateways **must** be defined for each subnet. For example, consider a cluster with subnets A, B, and C, as illustrated in the following figure:

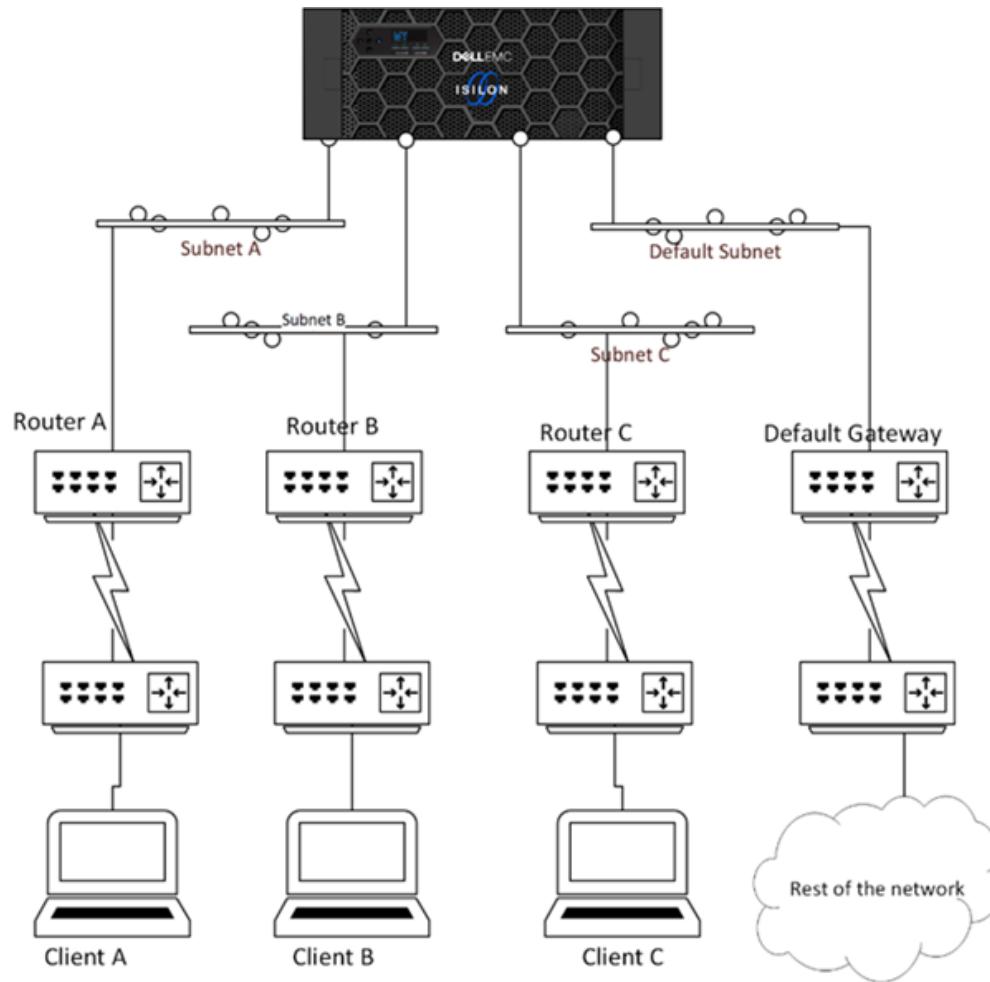


Figure 24 Source-Based Routing

In the example above, each gateway has a defined priority. If SBR is not configured, the highest priority gateway, i.e. gateway with the lowest value which is reachable, is used as the default route. Once SBR is enabled, when traffic arrives from a subnet that is not reachable via the default gateway, firewall rules are added. As OneFS is FreeBSD based, these are added through `ipfw`. In the example above, the following `ipfw` rules are provisioned:

```
If src-ip is in subnetA and dst-ip is not in (subnetA,B,C) set next-hop to gatewayA
If src-ip is in subnetB and dst-ip is not in (subnetA,B,C) set next-hop to gatewayB
If src-ip is in subnetC and dst-ip is not in (subnetA,B,C) set next-hop to gatewayC
```

The process of adding ipfw rules is stateless and essentially translates to per-subnet default routes. SBR is entirely dependent on the source IP address that is sending traffic to the cluster. If a session is initiated from the source subnet, the ipfw rule is created. The session must be initiated from the source subnet, otherwise the ipfw rule is not created. If the cluster has not received traffic that originated from a subnet that is not reachable via the default gateway, OneFS will transmit traffic it originates through the default gateway. Given how SBR creates per-subnet default routes, consider the following:

- A subnet setting of 0.0.0.0 is not supported and is severely problematic, as OneFS does not support RIP, RARP, or CDP.
- The default gateway is the path for all traffic intended for clients that are not on the local subnet and not covered by a routing table entry. Utilizing SBR does not negate the requirement for a default gateway, as SBR in effect overrides the default gateway, but not static routes.
- Static routes are an option when the cluster originates the traffic, and the route is not accessible via the default gateway. As mentioned above, static routes are prioritized over source-based routing rules.

### 11.1 Source-Based Routing and DNS

As discussed earlier in this paper, it's important to understand the path a specific session traverses throughout the network hierarchy. If SBR is configured on a cluster, this will also impact how the cluster creates sessions with other hosts, such as a DNS server.

In certain environments, PowerScale clusters with SBR enabled and multiple SmartConnect SIP (SSIP) addresses, have experienced excessive latency with DNS responses. As mentioned previously in this paper, keeping latency minimal is imperative through any transaction and the delayed DNS responses could impact DNS dependent workflows. The prior section explained how SBR dynamically assigns a gateway. In this instance, the route to the DNS server is changed as the session originated on a different interface based on the SSIP being addressed.

In order to prevent the additional latency with DNS responses, when SBR is enabled with multiple SSIP addresses, consider the following:

- If a single Access Zone is configured to have multiple SmartConnect zones and multiple subnets with SBR enabled, it is recommended to have a single SSIP.
- If a cluster is using a single DNS server, it is recommended to use a single SSIP.
- If multiple Access Zones are required within a single groupnet, then a single SSIP is recommended.

## 12 Isilon 6<sup>th</sup> generation 1 GbE interfaces

The Isilon Gen 6 platform provides a single 1 GbE interface per node. The original intent of the interface was to provide SSH access to the cluster. However, against best practice, many cluster configurations use these interfaces for data access, either intentionally or unintentionally.

The 1 GbE interface was not designed for data transfer, as the chipset does not support TCP Segmentation Offload (TSO) and Large Receive Offload (LRO), increasing the chances of dropped packets, port flapping, and other performance issues. TSO and LRO are available on the 10 GbE and 40GbE ports, which are optimized for data traffic.

---

**Note:** The 1 GbE interfaces on Isilon Gen 6 nodes should not be used for data transfer under any circumstances. During the initial cluster configuration, if possible, do not configure the 1 GbE interface. If the interface must be enabled, it should only be utilized for WebUI and SSH access.

---

As a best practice, the 1 GbE interfaces should not be configured. During the initial cluster configuration, only configure the 10 GbE and 40 GbE interfaces. If the 1 GbE interface must be utilized for the initial cluster configuration, after the cluster is configured, remove the 1 GbE interfaces from all IP pools.

If the 1 GbE interfaces must be configured, ensure the following:

- Configure the interfaces on a separate dedicated VLAN or subnet
- The 1 GbE interfaces should not share an IP pool with the 10 GbE and 40 GbE interfaces
- Configure a 'Gateway Priority' for the 1 GbE interfaces subnet. Set the priority to the highest value of all the cluster's subnets, ensuring the 1 GbE interfaces only carry traffic under scenarios where all of the other cluster interfaces are unavailable.
- Consider disabling SMB3 Multi-Channel (MC). During the client connection process, SMB3 MC provides all IP addresses across subnets, including the 1 GbE interfaces.

Cluster's utilizing the 1 GbE interfaces for anything other than WebUI and SSH access experience overall performance degradation. Protocol performance suffers if traffic is routed through the 1 GbE interface. SyncIQ reports random failures as it may communicate through the 1 GbE interfaces, even if a policy is configured not to use those interfaces. Additional performance degradation impacts may include authentication, SRS, CEE, anti-virus, and CloudPools.

Another factor to consider for leaving the 1 GbE interface unconfigured is that starting with OneFS release 8.0, the `bge` driver for the 1 GbE interface increments 'InputDiscards' for unwanted packets, such as multicast and spanning-tree packets. The 'InputDiscards' show as errors under the `netstat -in` command, complicating the cluster network monitoring.



## 13 Intelligent Platform Management Interface

OneFS 9.0 introduces support for the Intelligent Platform Module Interface (IPMI) on 6<sup>th</sup> generation Isilon nodes and PowerScale nodes. IPMI provides a dedicated management channel for lights-out management, external to OneFS. The supported IPMI features are power control and Serial over LAN (SoL).

Once IPMI is configured through OneFS, it can be accessed through the 'IPMITool', which is part of most Linux distributions, or other proprietary IPMI tools. On Gen 6 Isilon nodes, the 1 GbE interface becomes dual personality, continuing to support management, but now also supports IPMI. For PowerScale nodes, the 1 GbE iDRAC port provides support for IPMI.

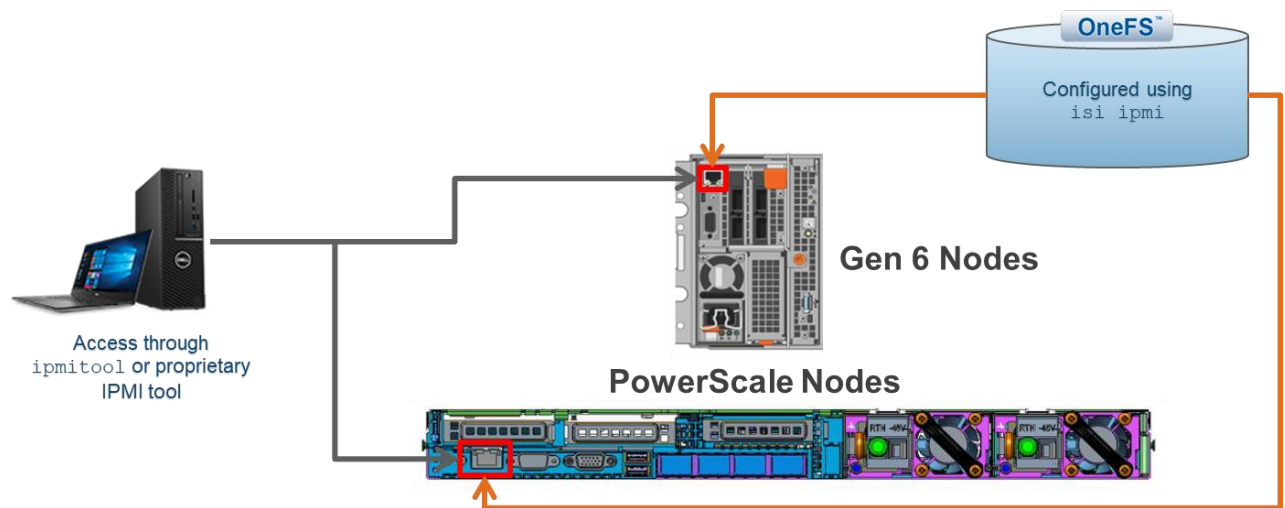


Figure 25 IPMI access for Gen 6 and PowerScale nodes

Before configuring IPMI, consider the following:

- Configuring the SoL feature on PowerScale nodes does require a node reboot to access and configure the serial communication settings.
- IPMI SoL is an alternative to physical serial cable access, but it is not a replacement for traditional SSH access to the cluster. Additionally, SoL is not available for initial cluster configuration as IPMI is disabled from the factory and must be enabled after initial cluster configuration.
- Gen 6 Isilon nodes require a Node Firmware Package at a minimum of v10.3.2.
- Configuring IPMI does require the `ISI_PRIV_IPMI` RBAC level, which is part of the Configuration privilege level.
- The IPMI IP address space cannot be on the same subnet as any of the cluster's front-end networks.
- On Gen 6 nodes, during an active SoL session, the physical serial port is disabled. Once the active SoL session is deactivated, the physical serial port becomes active again. Only a single SoL session is supported per node. On PowerScale nodes, the physical serial interface is disabled after the BIOS update explained in Section 13.2.1, Configure serial devices.
- A power reset may be issued through SoL, utilizing only the IPMI password for access.
- IPMI does not support VLAN tagging.
- As a security best practice, isolate IPMI traffic to a management only VLAN.
- At the initial login of the SoL session, the user is prompted for the OneFS CLI username and password. If the user logs out of the SoL session without logging out of the CLI, the CLI session remains active, allowing the next SoL session to enter the OneFS CLI without authenticating. As a security best practice, log out of the OneFS CLI session prior to logging out of the SoL session.

- Rebooting a node through SoL provides the full output of the entire OneFS shutdown and bootup sequence.

For releases prior to OneFS 9.0, IPMI is available for Gen 6 nodes as a manual configuration on each node. It is not officially supported, but it is also not prohibited, and it generally works. For OneFS 9.0, this process is an automated cluster configuration for all nodes within a cluster. If IPMI was configured on a release prior to OneFS 9.0, upgrading to 9.0 does not impact any existing IPMI configuration.

## 13.1 Configuring IPMI

From the OneFS CLI, to configure IPMI, perform the following:

1. Enable IPMI and configure the IPMI IP addresses with static or DHCP allocation:

```
isi ipmi settings modify --enabled=True --allocation type=[dhcp/static]
```

If IPMI is configured with static IP addresses, specify the IP range:

```
isi ipmi network modify --gateway=[Gateway IP] --prefixlen= --ranges=[IP Range]
```

---

**Note:** Assigning a specific IP address to a certain node may not be possible as IPs are assigned on a first-come basis. To confirm a specific nodes IP address, use the following command: `isi impi nodes list`

---

2. Enable the selected IPMI features:

```
isi ipmi features modify Power-Control --enabled=True
isi ipmi features modify SOL --enabled=True
```

---

**Note:** For PowerScale nodes, SoL requires additional configuration. Refer to Section 13.2, IPMI SoL on PowerScale nodes, after completing all the steps in this section.

---

3. Confirm the enabled IPMI features:

```
isi ipmi features list
```

4. Configure a single IPMI username and password for all nodes. The IPMI authentication is not part of any of the other OneFS authentication providers. It is only for IPMI access. Usernames up to 16 characters are supported. Passwords must be greater than 16 characters and less than 21 characters. Configure an IPMI username and password:

```
isi ipmi user modify --username=[Username] --set-password
```

Confirm a username is configured:

```
isi ipmi user view
```

## 13.2 IPMI SoL on PowerScale nodes

IPMI Serial over LAN (SoL) is enabled through the OneFS CLI, as explained in Section 13.1, Configuring IPMI. Once enabled, PowerScale nodes (F200/F600) require additional steps for activating SoL on each node in a cluster.

### 13.2.1 Configure serial devices

In order for IPMI SoL to function, the physical serial devices must be configured to support SoL on each node in the cluster.

**Note:** On PowerScale nodes, the physical serial port is no longer active after the BIOS configuration updates provided in this section.

The serial communication settings should be configured per the values specified in Table 11. For PowerScale nodes that do not have iDRAC configured, the serial communication settings are accessed during node boot up by selecting **F2**, entering the System BIOS, and selecting **Serial Communication**. For PowerScale nodes with iDRAC configured, the serial communication settings are found under **iDRAC > Configuration > BIOS Settings**. Update the serial communication settings per Table 11 and repeat for each node in the cluster.

Table 11 IPMI SoL configuration for PowerScale nodes

Field	Value
Serial Communication	On with console redirection to COM1
Serial Port Address	Serial Device 1 = COM2; Serial Device 2 = COM1
External Serial Connector	Serial Device 2
Failsafe Baud Rate	115200
Remote Terminal Type	VT100/VT200
Redirection After Boot	Enabled

### 13.2.2 iDRAC SoL permission

After the serial communication settings are configured, the IPMI username created in Section 13.1, Configuring IPMI, requires SoL permission through iDRAC. From the **iDRAC Settings** window, select the **Users** tab. Edit the IPMI username to have Serial over LAN enabled, as displayed in Figure 26 and Figure 27.

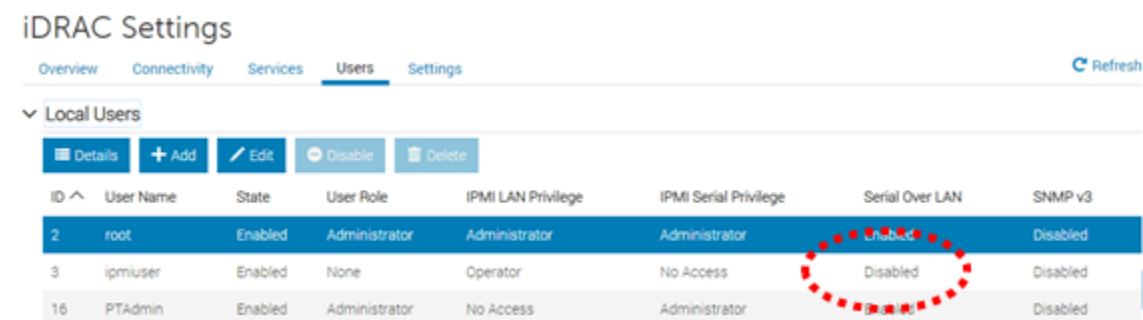
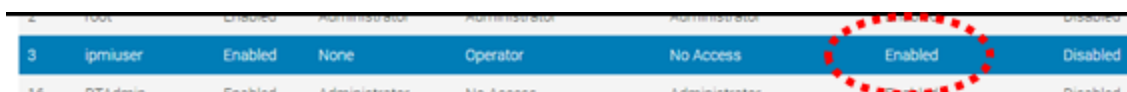


Figure 26 iDRAC Settings



ID	Username	Enabled	Password	Role	Access	Status
3	ipmiuser	Enabled	None	Operator	No Access	Enabled
16	OTAAdmin	Enabled	Administrator	No Access	Administrator	Disabled

Figure 27 IPMI user settings in iDRAC

The iDRAC settings must be updated on each PowerScale node in a cluster. Continue updating the iDRAC setting of each PowerScale node by accessing [https://<node\\_IPMI\\_IP\\_address>:443](https://<node_IPMI_IP_address>:443).

## 13.3 Accessing IPMI

Once IPMI is configured, it can be accessed through 3<sup>rd</sup> party proprietary tools, or most commonly through the 'IPMITool' command in Linux. The 'IPMITool' command is included in most Linux distributions.

To manage an Isilon or PowerScale node through IPMI, from a Linux endpoint, use the following syntax:

```
ipmitool -I lanplus -H [Node IP] -U [Username] -L OPERATOR -P [password] power
[status, on, off, cycle, reset]
```

```
ipmitool -I lanplus -H [Node IP] -U [Username] -L OPERATOR -P [password] sol
[info, activate, deactivate]
```

To exit an active SoL session, enter the following (including the period): ~.

To check if a node has an active SoL connection, enter the following: `ipmitool channel info 1`

## 13.4 Troubleshooting IPMI

The IPMI log file is stored at `/var/log/isi_ipmi_mgmt_d.log`

## 14 IPv6

Although Internet Protocol version 4 (IPv4) is the most common version of IP today, Internet Protocol version 6 (IPv6) is the newest version and ultimately replaces IPv4. IPv4 addresses were completely allocated to specific geographic regions in 2011. IPv6 uses 128-bit addresses supporting 340 undecillion addresses. For those unfamiliar with an undecillion, this translates to 340 times 10 to the 36th power possible IP addresses.

### 14.1 Why IPv6?

IPv6 brings innovation and takes connectivity to a new level with enhanced user experiences.

#### 14.1.1 Security

IPv6 supports IPSEC inherently with encryption and integrity checks. Additionally, the Secure Neighbor Discovery (SEND) protocol provides cryptographic confirmation of host identity, minimizing hostname-based attacks like Address Resolution Protocol (ARP) poisoning, leading to devices placing more trust in connections.

#### 14.1.2 Efficiency

IPv6's large address space means many devices no longer require NAT translation as previously with IPv4, making routers far more efficient. Overall data transmission is faster and simplified as the need for checking packet integrity is eliminated.

#### 14.1.3 Multicast

IPv6 supports multicast rather than broadcast, allowing media streams to be sent to multiple destinations simultaneously leading to bandwidth savings.

#### 14.1.4 Quality of Service

Quality of Service (QoS) implementation is simplified in IPv6 with a new packet header. The IPv6 header contains a new field, Flow Label, which identifies packets belonging to the same flow. The Flow Label associates packets from a specific host and head to a particular destination.

## 14.2 IPv6 addressing

IPv6's address structure is defined by the IETF as part of RFC 3513 and provides many of the advantages discussed above over IPv4. At first glance, it is evident an IPv6 address, looks nothing like an IPv4 address. IPv4 addresses are composed of four numerical octets, ranging from 0 to 255, separated by periods, forming a 32-bit address. IPv6 addresses are 128 bits and consisting of a series of eight segments, separated by a colon. Each segment is a 4-character hexadecimal number, ranging from 0000 to FFFF, representing 16 bits each, totaling to the 128 bits.

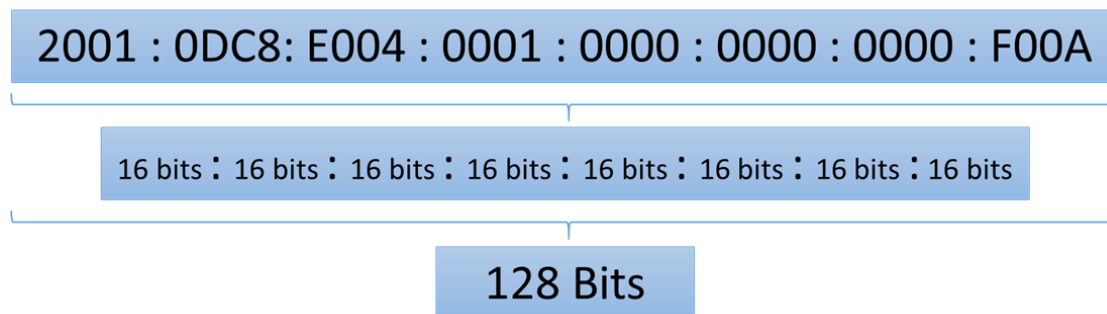


Figure 28 IPv6 address

For display purposes, an IPv6 address may be presented without leading zeros. For example, an IPv6 address of 2001 : 0DC8: E004 : 0001 : 0000 : 0000 : 0000 : F00A could be displayed as 2001 : DC8: E004 : 1 : 0 : 0 : 0 : F00A.

The address may be further reduced by removing consecutive fields of zeros and replacing with a double-colon. The double-colon can only be used once in an address. The address above becomes 2001 : DC8 : E004 : 1 :: F00A.

IPv6 offers the following address types:

- Unicast: one-to-one – Single Address to Single Interface
- Anycast: one-to-nearest – Assigned to a group of interfaces, with packets being delivered only to a single (nearest) interface
- Multicast: one-to-many – Assigned to a group of interfaces and is typically delivered across multiple hosts.

An IPv6 Unicast address is composed of the Global Routing Prefix, Subnet ID, and the Interface Identifier. The Global Routing Prefix is the network ID or prefix of the address for routing. The Subnet ID is similar to the netmask in IPv4 but is not part of the IP address in IPv6. Finally, the Interface ID is a unique identifier for a particular interface. For Ethernet networks, the Ethernet MAC address (48 bits) may be used for the Interface Identifier, by inserting 16 additional bits, forming what is referred to as an EUI-64 address.

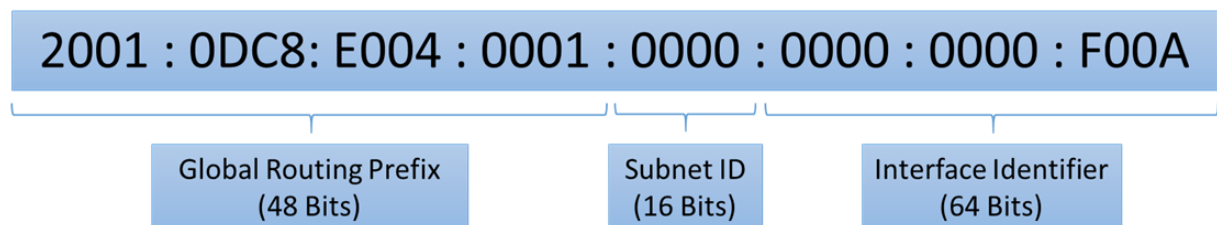


Figure 29 IPv6 Unicast address format

## 14.3 IPv6 header

An IPv6 header is simplified in comparison to IPv4, minimizing complexity and making the header easier to process for all devices. Efficiency was one of the focus points with IPv6 from the onset, which is brought to light with the faster processing of IPv6 headers. The figure below displays the format of an IPv6 header.

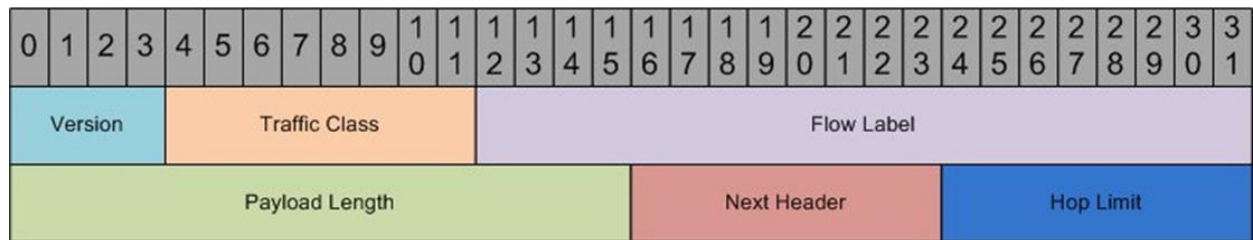


Figure 30 IPv6 header

The table below defines the fields of an IPv6 header.

Table 12 IPv6 header fields

Field	Description	Length
Version	Specifies if the packet is IPv4 or IPv6	4 Bits
Traffic Class	Similar to an IPv4 Type of Service field and includes support for Differentiated Services Code Point (DSCP) providing congestion control.	8 Bits
Flow Label	Provides the ability to track certain traffic flows at the network layer for QoS management.	20 Bits
Payload Length	Similar to the IPv4 'Length' field – Provides length of the data portion	16 Bits
Next Header	Similar to the IPv4 'Protocol' field – Provides what to expect after the basic header, including options for a TCP or UDP header	8 Bits
Hop Limit	Similar to the IPv4 'Time to Live' field – Provides the maximum number of hops	8 Bits

## 14.4 IPv6 to IPv4 translation

Connecting IPv6 and IPv4 remains a challenge with the slow migration to IPv6 and support for legacy devices requiring IPv4. The three top options available to facilitate IPv4 and IPv6 communication are dual-stack networks, tunneling, and translation.

For Service Providers to deliver IPv6, they utilize translation technologies. The two major translation technologies are the Network Address Translation IPv6 to IPv4 (NAT64) and Stateless IP/ICMP Translation (SIIT). NAT64 is similar to the IPv4 Network Address Translation but is specific to IPv6. SIIT is capable of replacing IPv4 and IPv6 as part of the translation.

## 14.5 Configuring OneFS for IPv6

Implementing IPv6 on a PowerScale cluster is a simple process as IPv6 and IPv4 are supported as a dual stack. In order to provision an IPv6 subnet with PowerScale, follow these steps:

1. Select an existing Groupnet or create a new one
2. Enter DNS servers and add a Subnet
3. Create Subnet by selecting IPv6 in the IP Family
4. Create a network address pool and assign interfaces



## 15 Network troubleshooting

This section provides steps for assessing and troubleshooting network issues with generally available utilities.

### 15.1 Netstat

Netstat, short for network statistics, is a utility built into most Windows and Linux clients. It provides an array of statistics on current ports, routing, IP stats for transport layer protocols, and serves as a forensic tool to link processes with network performance while digging deeper into the current network status. Netstat bundles several actions into a single command with different options available. As Netstat is multi-platform, the syntax is similar across platforms with slight variations.

#### 15.1.1 Netstat

In its standard form without any arguments, netstat provides an overview of the current network status broken out by each connection or socket. Each column displays the following:

- **Proto:** Protocol of the active connection. The protocol could be TCP or UDP and has a '4' or '6' associated specifying if it is IPv4 or IPv6, respectively.
- **Recv-Q and Send-Q:** Value of the receiving and sending queue in bytes. Non-zero values specify the number of bytes in the queue that are awaiting to be processed. The preferred value is zero. If several connections have non-zero values, this implies something is causing processing to be delayed.
- **Local Address and Foreign Address:** Lists the hosts and ports the sockets are connected with. Some of these are local connections to the host itself.
- **State:** Displays the current state of the TCP connection, based on the TCP protocol. As UDP is a stateless protocol, the 'State' column will be blank for UDP connections. The most common TCP states include:
  - **Listen:** Waiting for an external device to establish a connection
  - **Established:** Ready for communication on this connection
  - **Close Wait:** The remote machine has closed the connection, but the local device has not closed the connection yet.
  - **Time Wait:** The local machine is waiting for a period of time after sending an ACK to close a connection.

For more information about the states of a TCP connection, see RFC 793.

```
tme-sandbox-1# netstat
Active Internet connections
Proto Recv-Q Send-Q Local Address           Foreign Address         (state)
tcp4    0      0 10.245.109.81.49318     10.245.109.165.12228    TIME_WAIT
tcp4    0      0 tme-sandbox-1.efs      tme-sandbox-1.9101     TIME_WAIT
tcp4    0      0 10.245.109.81.57047     10.245.109.165.12228    TIME_WAIT
tcp4    0      0 10.245.109.81.49868     10.245.109.165.12228    TIME_WAIT
tcp4    0      0 10.245.109.81.32759     10.245.109.165.12228    TIME_WAIT
tcp4    0      0 10.245.109.81.55357     10.245.109.165.12228    TIME_WAIT
tcp4    0      0 10.245.109.81.44141     10.245.109.165.12228    TIME_WAIT
tcp4    0      0 10.245.109.81.23947     10.245.109.165.12228    TIME_WAIT
```

Figure 31 Netstat

Netstat reveals a lot of information about the current status of network connections, and it also provides information for a more thorough forensic analysis. While reviewing the output from netstat, some of the scenarios can be generalized, like the following:

- Recv-Q has a value greater than zero but is in a 'Close Wait' state. This indicates that these sockets should be torn down but are hanging. If several sockets are in this state, it could imply the application is having difficulty tearing down the connection and may warrant additional investigation.
- Connections that have localhost as the 'Local' and 'Foreign' address denote an internal process using the TCP stack to communicate. These connections are not concerning and are standard practice.

### 15.1.2 netstat -s -p tcp

Netstat offers several options, but the '-s' provides statistics by protocol and '-p' displays the net to media tables. These options reveal current health and the 'tcp' limits it to the TCP protocol. Below, is a sample output of this command with the areas to examine highlighted in red.

```
tme-sandbox-1# netstat -s -p tcp | more
tcp:
  235829612 packets sent
    120878277 data packets (268468965468 bytes)
    249379 data packets (336827964 bytes) retransmitted
    1418 data packets unnecessarily retransmitted
    0 resends initiated by MTU discovery
    108808793 ack-only packets (32714420 delayed)
    0 URG only packets
    0 window probe packets
    2286382 window update packets
    3606781 control packets
  323541220 packets received
    150675311 acks (for 268424631560 bytes)
    1356119 duplicate acks
    0 acks for unsent data
    224269521 packets (250631325650 bytes) received in-sequence
    83094 completely duplicate packets (2312228 bytes)
    4841 old duplicate packets
    133 packets with some dup. data (11814 bytes duped)
    638885 out-of-order packets (896803550 bytes)
    0 packets (0 bytes) or data after window
    0 window probes
    1550748 window update packets
    338 packets received after close
    0 discarded for bad checksums
    0 discarded for bad header offset fields
    0 discarded because packet too short
    0 discarded due to memory problems
  1539694 connection requests
  2095764 connection accepts
  0 bad connection attempts
```

Figure 32 netstat -s -p tcp

The fields highlighted in red above must be reviewed as a ratio of the total packets that are transmitted and received as a percentage. Additionally, these statistics should be monitored for sudden increments. As a guideline, under 1% is not concerning but this also depends on the workload. The fields highlighted above provide the following:

- Retransmitted Packets: Packets that are retransmitted consume network bandwidth and could be the reason for further investigation. However, examining the percentage is critical. In this case, 249379 out of 235829612 were retransmitted, which is 0.105%.
- Duplicate Acknowledgements: High latency between endpoints may lead to duplicate acknowledgments, but the ratio must be examined. In this case, it is 0.419%. This number varies depending on the workload.
- Out of Order Packets: Out of order packets are placed in order by TCP before presenting to the application layer, which impacts the CPU and overall stack performance as the additional effort is

involved in analyzing the packets. Performance is impacted the most when packets arrive out of order with a significant time gap, or a number of packets are out of order. The ratio, in this case, is 0.197%, which is negligible.

### 15.1.3 netstat -i

Another option for netstat is the `-i` option, which is the interface display, listing cumulative statistics for total packets transferred, errors, MTU, and collisions by the interface. As netstat `-i` lists all available interfaces, the back-end, and front-end interfaces are displayed. A sample output of netstat `-i` is shown with the `-h` option, making it easier to interpret, in the following figure:

```
tme-sandbox-1# netstat -i -h
Name      Mtu Network      Address      Ipkts Ierrs Idrop      opkts oerrs coll
bxe0      1.5K <Link#1>      00:0e:1e:5b:3f:90 2.9M      0      0      147      0      0
bxe0      - 192.168.192.0 192.168.200.240 702K      -      -      120      -      -
bxe1      1.5K <Link#2>      00:0e:1e:5b:3f:92 2.9M      0      0      147      0      0
bxe1      - 192.168.192.0 192.168.200.243 702K      -      -      120      -      -
igb0      1.5K <Link#3>      00:1e:67:ef:fc:0c 267M      0      0      113M     0      0
igb0      - 10.245.109.0 10.245.109.81 206M      -      -      217M     -      -
igb1      1.5K <Link#4>      00:1e:67:ef:fc:0d      0      0      0      0      0      0
lo0       16K <Link#5>      ::1              60M      0      0      60M      0      0
lo0      - localhost      ::1              11M      -      -      11M      -      -
lo0      - fe80::1%lo0    fe80::1%lo0      0      -      -      0      -      -
lo0      - your-net      localhost        32M      -      -      45M      -      -
lo0      - 127.42.0.0     127.42.0.1      752K      -      -      752K     -      -
ib0       4.0K <Link#6>      00:00:00:48:fe:80:      0      0      0      0      3.3M     0
ib1       2.0K <Link#7>      00:00:00:49:fe:80: 132M      0      0      84M      15      0
ib1      - 192.168.81.0  tme-sandbox-1  88M      -      -      81M      -      -
```

Figure 33 netstat -i

From the output above, netstat `-i`, lists the following columns:

- **Name:** Network Interface Card (NIC) name. Loopback interfaces are listed as 'lo0,' and 'ib' specifies InfiniBand.
- **MTU:** Lists the MTU specified for the interface.
- **Network:** Specifies the network associated with the interface.
- **Address:** MAC address of the interface.
- **Ipkts:** Input packets are the total number of packets received by this interface.
- **Ierrs:** Input errors are the number of errors reported by the interface when processing the 'Ipkts.' These errors include malformed packets, buffer space limitation, checksum errors, errors generated by media, and resource limitation errors. Media errors are errors specific to the physical layer, such as the NIC, connection, cabling, or switch port. Resource limitation errors are generated at peak traffic when interface resources are exceeded by usage.
- **Idrop:** Input drops are the number of packets that were received, but not processed and consequently dropped on the wire. Dropped packets typically occur during heavy load.
- **Opkts:** Output packets are the total number of packets transmitted by this interface
- **Oerrs:** Output errors are the number of errors reported by the interface when processing the 'Opkts.' Examples of errors include the output queue reaching limits or an issue with the host.
- **Coll:** Collisions are the number of packet collisions that are reported. Collisions typically occur during a duplex mismatch or during high network utilization.

In general, errors and dropped packets require closer examination. However, as noted in the previous netstat section, the percentage of errors and dropped packets are the main factor. The following are some of the points to consider for further analysis:

- 'Ierrs' should typically be less than 1% of the total 'Ipkts.' If greater than 1%, check 'netstat -m' for buffer issues and consider increasing the receive buffers. Prior to implementing changes on a

production system, buffer changes should be tested in a lab environment. Refer to the PowerScale Network Stack Tuning Section for additional details.

- 'Oerrs' should typically be less than 1% of the total 'Opkts.' If greater than 1%, it could be a result of network saturation, otherwise consider increasing the send queue size.
- The ratio of 'Coll' to 'Opkts,' should typically be less than 10%. If greater than 10%, it could be a result of high network utilization.

### 15.1.4 netstat -m

The netstat -m option displays the current status of network memory requests as mbuf clusters. Netstat -m is a powerful option for a complete forensic analysis when one of the other netstat commands mentioned above raises concern. If mbufs are exhausted, the node cannot accept any additional network traffic.

```
tme-sandbox-1# netstat -m
31228/9287/40515 mbufs in use (current/cache/total)
25045/1565/26610/0 mbuf clusters in use (current/cache/total/max)
25045/1520 mbuf+clusters out of packet secondary zone in use (current/cache)
4598/67/4665/0 4k (page size) jumbo clusters in use (current/cache/total/max)
0/0/0/451826 9k jumbo clusters in use (current/cache/total/max)
0/0/0/254152 16k jumbo clusters in use (current/cache/total/max)
76289k/5719k/82008k bytes allocated to network (current/cache/total)
0/0/0 requests for mbufs denied (mbufs/clusters/mbuf+clusters)
0/0/0 requests for mbufs delayed (mbufs/clusters/mbuf+clusters)
0/0/0 requests for jumbo clusters delayed (4k/9k/16k)
0/0/0 requests for jumbo clusters denied (4k/9k/16k)
0 requests for sbufs denied
0 requests for sbufs delayed
3 requests for i/o initiated by sendfile
```

Figure 34 netstat -m

The netstat -m output provides information in regard to available and used mbufs. The area highlighted in red, confirms if any memory requests have been denied. In the example above, a quick glance at this area reveals that no requests have been denied.

For more information on netstat options, visit the FreeBSD manual netstat page at <https://www.freebsd.org/cgi/man.cgi?query=netstat&manpath=SuSE+Linux/i386+11.3>

## 15.2 InsightIQ external network errors

PowerScale InsightIQ reports external network errors under the “Performance Reporting” tab when the “Network Performance Report” is selected. A sample of this output is displayed in the following figure:

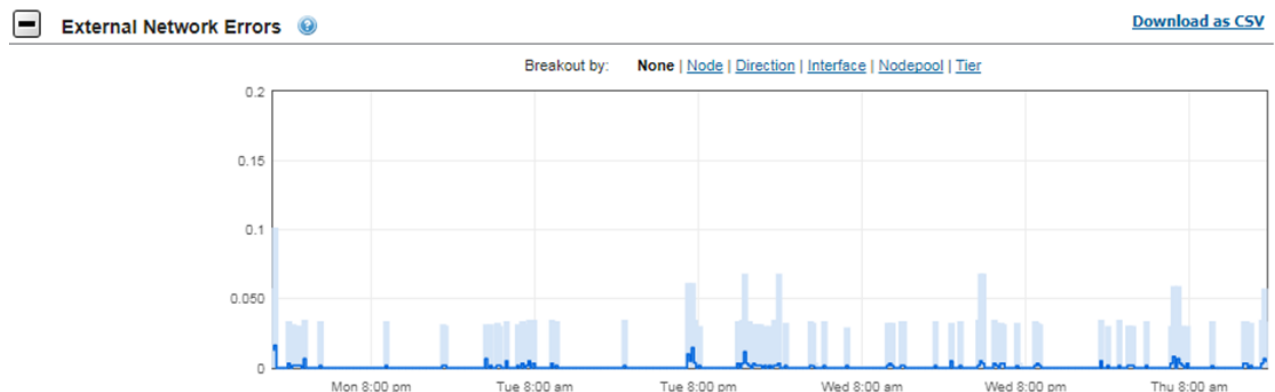


Figure 35 InsightIQ network errors



InsightIQ gathers network errors using the output from 'netstat -i' on external interfaces only. The total of the 'lerrs' and 'Oerrs' is combined and displayed in the graph. Refer to the previous section for interpreting the output from 'netstat -i.'

In order to find the exact interface errors, sort the graph by 'Node,' 'Direction,' and 'Interface,' as shown in the following figures:

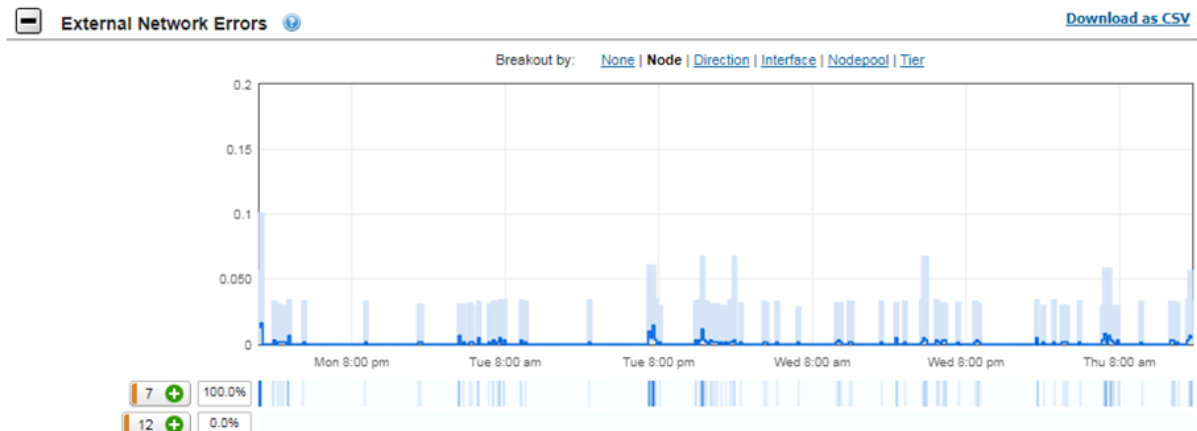


Figure 36 InsightIQ external network errors by node

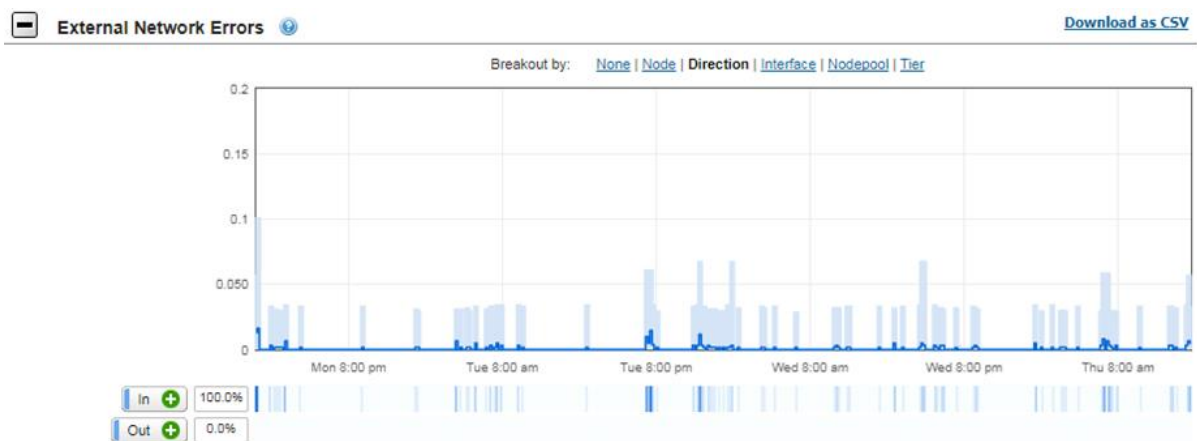


Figure 37 InsightIQ external network errors by direction

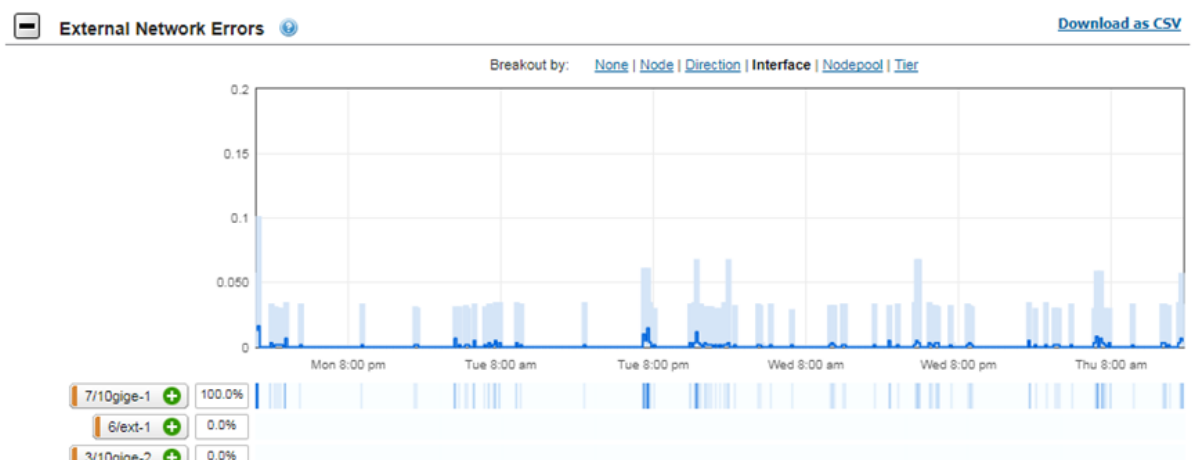


Figure 38 InsightIQ external network errors by interface

From the figures above, it is concluded that the external network errors reside on the interface '7/10gige-1' of Node 7, on the input or receive side. Further analysis must be performed on this interface to conclude the root cause. Refer to the 'netstat -i' section in this paper for the next troubleshooting steps.

## 15.3 DNS

DNS or Domain Name Service resolves hostnames to IP addresses. Most enterprises have a local DNS to resolve hostnames managed by them, and then a public internet DNS resolves external hostnames. Troubleshooting DNS is performed with the utilities, 'nslookup' or 'dig.' Both provide similar information; however, 'dig' is more detailed. In this section, the usage of 'dig' is explored.

```
dig dell.com

; <<>> DiG 9.8.3-P1 <<>> dell.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33189
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 6, ADDITIONAL: 6

;; QUESTION SECTION:
;dell.com.                IN      A

;; ANSWER SECTION:
dell.com.                 554     IN      A      143.166.135.105
dell.com.                 554     IN      A      143.166.147.101

;; AUTHORITY SECTION:
dell.com.                 61      IN      NS      ns3.us.dell.com.
dell.com.                 61      IN      NS      ns6.us.dell.com.
dell.com.                 61      IN      NS      ns4.us.dell.com.
dell.com.                 61      IN      NS      ns2.us.dell.com.
dell.com.                 61      IN      NS      ns1.us.dell.com.
dell.com.                 61      IN      NS      ns5.us.dell.com.

;; ADDITIONAL SECTION:
ns6.us.dell.com.          19      IN      A      143.166.224.235
ns5.us.dell.com.          172350  IN      A      143.166.83.13
ns3.us.dell.com.          172327  IN      A      143.166.224.3
ns2.us.dell.com.          172327  IN      A      143.166.82.252
ns1.us.dell.com.          172327  IN      A      143.166.82.251
ns4.us.dell.com.          172350  IN      A      143.166.224.11

;; Query time: 26 msec
;; SERVER: 192.168.50.1#53(192.168.50.1)
;; WHEN: Mon Oct 2 10:45:03 2017
;; MSG SIZE rcvd: 265
```

Figure 39 dig dell.com

The 'dig' command displays results in the following sections:

- Header: The Header provides the version of 'dig', options the 'dig' command used and the flags that are displayed.
- Question Section: The Question Section displays the original input provided to the command. In the case above, dell.com was queried. The default is to query the DNS A record. Other options are available for querying MX and NS records.
- Answer Section: The Answer Section is the output received by dig from the DNS server queried.
- Authority Section: The Authority Section lists the available Name Servers of dell.com. They have the authority to respond to this query.

- Additional Section: The Additional Section resolves the hostnames from the Authority Section to IP addresses.
- Stats Section: The footer at the end of the query is referred to as the Stats Section. It displays the when, where, and time the query consumed.

Dig supports an array of options. The most common options include a reverse look-up using 'dig -x [IP address]' to find a host name. The other is to specify a DNS server to query using 'dig @[dns server] [hostname].'

For the complete list of dig options, please refer to the FreeBSD manual page:

<https://www.freebsd.org/cgi/man.cgi?query=dig&sektion=1&manpath=FreeBSD%209.1-RELEASE>

## A Supported network optics and transceivers

For information about the optics and transceivers supported by PowerScale nodes, refer to the [PowerScale Supportability and Compatibility Guide](#).



## B Technical support and resources

[Dell.com/support](https://dell.com/support) is focused on meeting customer needs with proven services and support.

[Storage technical documents and videos](#) provide expertise that helps to ensure customer success on Dell EMC storage platforms.

### B.1 Related resources

Dell Networking Solutions: <https://infohub.delltechnologies.com/t/networking-solutions-57/>

Dell EMC Networking with Isilon Front-End Deployment and Best Practices:  
<https://infohub.delltechnologies.com/t/dell-emc-networking-with-isilon-front-end-deployment-and-best-practices-1/>

Isilon Networking Front-End Deployment - Networking Configuration - Videos:  
<https://infohub.delltechnologies.com/t/isilon-networking-front-end-deployment-part-1-networking-configuration-video-1/>

IEEE Standards: <https://standards.ieee.org/findstds/standard/802.1AX-2008.html>

SMBv3 Multi-Channel:  
<https://blogs.technet.microsoft.com/josebda/2012/06/28/the-basics-of-smb-multichannel-a-feature-of-windows-server-2012-and-smb-3-0/>

PowerScale OneFS Documentation:

- [OneFS 9.0.0.0 Documentation - PowerScale Info Hub](#)
- [OneFS 9.0.0.0 Web Administration Guide](#)
- [OneFS 9.0.0.0 CLI Administration Guide](#)
- [OneFS 9.0.0.0 API Reference](#)
- [OneFS 9.0.0.0 Security Configuration Guide](#)

RFCs:

- <http://www.faqs.org/rfcs/rfc1812.html>
- <http://www.faqs.org/rfcs/rfc1122.html>
- <http://www.faqs.org/rfcs/rfc1123.html>
- <https://tools.ietf.org/html/rfc3971>
- <https://tools.ietf.org/html/rfc792>
- <https://tools.ietf.org/html/rfc3513>