ethernet alliance

Data Center Bridging Plugfest

Version 1.0, August 2009

Contributors:

Eddie Tan, Cisco
Gary Gumanow, Dell
Kirt Gillum, Dell
Joy Jiang, Finisar
Dan Daly, Fulcrum Microsystems
Shawn Dutton, NetApp
Manoj Wadekar, QLogic
Mikkel Hagen, UNH-IOL
Sean Hull, Spirent
Sunil Ahluwalia, Intel
Paul Doherty, Intel

Ethernet Alliance | 3855 SW 153rd Drive | Beaverton, OR 97006
www.ethernetalliance.org

Table of Contents

1 Introduction & Background

In May 2009, the Ethernet Alliance sponsored an interoperability event where eight Ethernet vendors came together at the University of New Hampshire Interoperability Lab (UNH-IOL) for the first of several closed door interoperability test events, also known as "plugfests". The purpose for this event was to test the interoperability of several Data Center Bridging (DCB) technologies, which are a collection of various standards efforts that are in progress inside of IEEE 802.1™. The purpose of these extensions is to enhance the capabilities of Ethernet and is a part of the many efforts of the Ethernet in the Data Center technologies. The term DCB describes enhancements to the existing Ethernet technology that enables convergence of various applications in data centers (LAN, SAN, and HPC) onto a single interconnect technology. As a subcommittee within the Ethernet Alliance the Ethernet in the Data Center subcommittee focuses on education and testing of Ethernet products in the data center.

The objectives of the Ethernet in the Data Center subcommittee include:

- Act as a reference and resource for the industry for both existing and emerging data center-focused Ethernet technologies
- Identify additional areas of technical work to ensure Ethernet leadership as the fabric in the data center
- Sponsor interoperability events and plug-fests for various emerging technologies through third party test facilities
- Align with IEEE 802® task force members and T11 members to promote emerging Ethernet standards for data center networking

1.1 DCB Plugfest Objectives

The objective of this first plugfest was to expedite the debugging and vendor/product interoperability between the participating companies and their products. The technologies being tested were Data Center Bridging Exchange (DCBX), Priority-base Flow Control (PFC) and Enhanced Transmission Selection (ETS). Vendors participating in this plugfest are shown in Table 1.

Table 1 - Vendors Participating

Vendor	Model	Product Category	Hardware Revision	Software/ Firmware	Technology Tested
Cisco	Nexus 5010	Switch	1	4.1(3)N1(0.86)	DCBX,PFC, ETS
Dell	PS Series Array	SAN Arrays	Prototype	Prototype	DCBX, PFC

Vendor	Model	Product Category	Hardware Revision	Software/ Firmware	Technology Tested
Finisar	Xgig Medusa test tools	Analyzer and Traffic Gen	Xgig 10GbE Blade	Xgig 4.5 Maestro 4.7 MLTT 3.0	DCBX, PFC, ETS
Fulcrum Microsystems	Monaco DCB Switch Reference Design	Switch	FM4224-A3	API 2.5.2(Testpoint)	DCBX, PFC, ETS
Intel	Intel® Ethernet Server Adapter X520-DA2	LAN Adapters	B0	VC2357	DCBX, PFC, ETS
NetApp	FAS3000 Series	NAS/FCoE Array	Q FW-5.1.1/MPI FW-1.35.0	7.3.2	DCBX, PFC, ETS
QLogic	QLE8152 CNA	LAN Adapters	A1	5.01.01 Stor miniport 9.1.8.17 A2	DCBX, PFC, ETS
Spirent TestCenter	Performance & Compliance Test Suite	MSA-2001B	SW-3.30.1721	Spirent TestCenter	DCBX, PFC, ETS

1.1.1 Brief description of the standard and technology

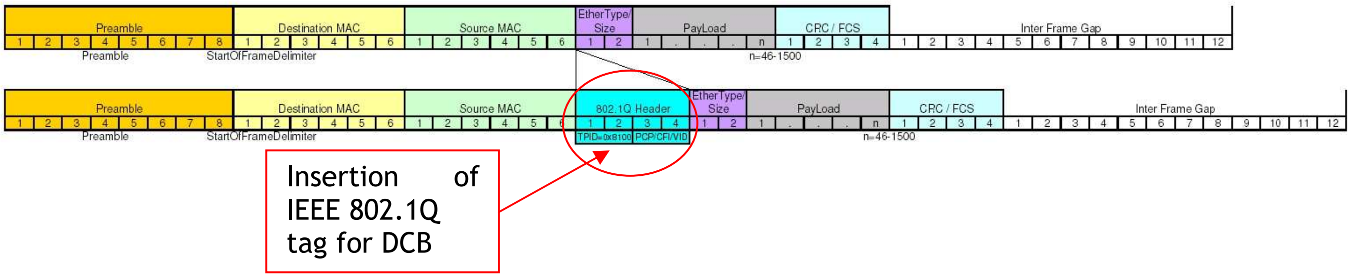
With data centers becoming bigger and more complex, managing different interconnect technologies for traffic from each application is becoming cost- and resource-intensive. Data centers deploy different networks based on distinct interconnect technologies to transport different traffic from different applications; for example, storage traffic is transported over TCP/IP-based iSCSI SAN, Fibre Channel-based SAN, or InfiniBand. Client-server application traffic is handled by an Ethernet-based LAN, while server-to-server IPC may be supported over one of various interconnects such as InfiniBand or Myrinet. A typical server in a high-performance data center has multiple interfaces (Ethernet, FC, and InfiniBand) to allow it to be connected to the various disparate networks.

One of the motivating factors for convergence is the consolidation of servers brought about by the advent of blade servers. Today, blade servers must accommodate backplane designs supporting multiple interconnecting technologies. However, using a single interconnect technology such as Ethernet can simplify backplane designs, thereby reducing overall costs and power consumption.

However, current standards-based Ethernet networks have not met the requirements to meet the demands of storage and high-performance computing applications. To understand the current limitations and enhancements required, one needs to know about the current state of Ethernet, the high-level requirements of a converged data center network, enhancements needed to current Ethernet, and relevant standards.

Through its simplicity and adaptability Ethernet has evolved to become THE foundational technology for a truly unified fabric. As a result the DCB standards were created to provide standardized enhancements to Ethernet to support converged fabric capabilities, but using industry standards as opposed to proprietary approaches.

Data Center Bridging enabled switches, NICs, HBAs, and storage targets, insert an IEEE 802.1Q tag into the Ethernet frame so that they can provide the necessary bandwidth for specific applications. This tag is read by the DCB enabled device for priority and bandwidth decisions. Figure 1 shows an Ethernet frame before and after the DCB tag is inserted.



Insertion of IEEE 802.1Q tag for DCB

Figure 1 - Ethernet frame with DCB tag inserted

1.2 What problems does DCB solve?

Ethernet has offered mechanisms to “prioritize” certain classes of traffic for several years with IEEE 802.1p; however, it does not allow for the control mechanisms to throttle individual classes of Ethernet frames. In other words, Ethernet can prioritize frames tagged as Voice over IP (VoIP) over LAN traffic, but if bandwidth from VoIP and LAN saturates the entire Ethernet connection, frames will be dropped. The dropped packet will then be retransmitted.

Ethernet does not currently have adequate facilities to control and manage the allocation of network bandwidth to different network traffic sources and/or types (traffic differentiation) or to allow the management capabilities to efficiently and fairly prioritize bandwidth utilization across these sources and traffic types. Lacking these complete capabilities, data center managers must either over provision network bandwidth for peak loads, accept customer complaints during these periods, or manage traffic prioritization at the source side by limiting the amount of non-priority traffic entering the network.

Overcoming these limitations is the key to enabling Ethernet as the foundation for true converged data center networks supporting the LAN, storage, and inter-processor communications. Ethernet has been successful in evolutionary enhancements due to its ability to be backward compatible and allowing plug-and-play deployment.

DCB needs to be able to interoperate with traditional Ethernet devices that do not have DCB capabilities. This plug-and-play functionality is provided to DCB devices by a protocol called DCB Capability eXchange Protocol (DCBX). This protocol is defined in IEEE 802.1Qaz Task Force. It provides ability for Ethernet devices (bridges, end stations) to detect DCB capability of the peer device. It also allows configuration distribution from one node to another. This simplifies management of DCB nodes significantly. The DCBX protocol uses LLDP (defined by IEEE Std. 802.1AB™-2005) services for exchanging DCB capabilities.

2 Testing

2.1 Test Methodology

Data Center Bridging introduces two low level features to Ethernet that enable a single Ethernet network support these disparate traffic types:

- Priority-base Flow Control
- Enhanced Transmission Selection

The configuration management of these features is defined in DCBX.

The goal of the Ethernet Alliance's testing is to demonstrate and verified these three features in a multivendor environment.

The baseline version of the standards that all vendors agreed to use can be found here:

http://www.ieee802.org/1/files/public/docs2008/dcb-baseline-contributions-1108-v1.01.pdf

The individual documents for each technology can be found at the following links:

- IEEE P802.1Qbb: Priority-based Flow Control: http://www.ieee802.org/1/files/public/docs2008/bb-pelissier-pfc-proposal-0508.pdf
- IEEE P802.1Qaz: Enhanced Transmission Selection (aka Priority Groups): http://www.ieee802.org/1/files/public/docs2008/az-wadekar-ets-proposal-0608-v1.01.pdf
- IEEE P802.1Qaz: DCB Capability Exchange Protocol (DCBX): http://www.ieee802.org/1/files/public/docs2008/az-wadekar-dcbx-capability-exchange-discovery-protocol-1108-v1.01.pdf

2.2 What we tested

2.2.1 Test Track 1: PFC Test

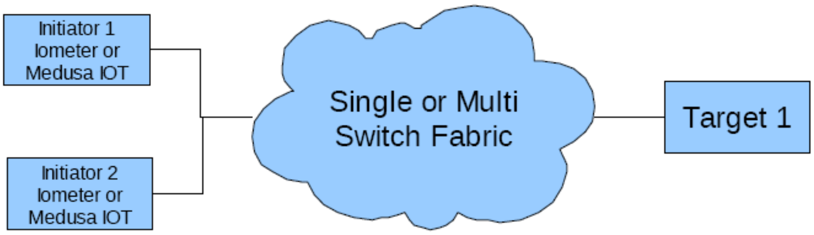


Figure 2 – PFC Test Configuration

Procedure: Link partners will transmit/receive priority pause as appropriate. Link partners in this test track must support mapping (at least two) traffic classes to VLAN priorities as well as enabling PFC on at least one of those VLAN priorities. ETS and DCBX support is not required (Unless defined in the test setup). In this test track multiple traffic class flows are used. Accepted flows and tools include:

- LAN (iperf/netperf/IxChariot/IxChariot/Spirent TestCenter Virtual), SAN (IOMeter/IOZone/MedusaTools/Storage Targets) and Finisar Xgig/Jammer. At least one of the following test setups needs to be supported for device under test.

1 ***Switch to Switch PFC Interop***: Using the test configuration shown, transmit multiple traffic classes from the first traffic generator to the second traffic generator.

2 ***Switch to End Device PFC Interop***: Using the test configuration shown, transmit multiple traffic classes from the traffic generator to the end device. If transmitting at line rate does not cause PFC to be generated, insert a test device to pause the transmit of the end device.

3 ***End Device to Switch PFC Interop:*** Using the test configuration shown, transmit multiple traffic classes from the end device to the traffic generator storage target (For FCoE based storage targets DCBX needs to be enabled on the devices in the network).

4 ***End Device to End Device PFC Interop***: Using the test configuration shown, transmit multiple traffic classes from the first end device to the second end device. If transmitting at line rate does not cause PFC to be generated, insert a test device to pause the transmit of the second end device.

Observable Results:

- Verified that the second switch transmits a valid PFC frame to the first switch and the first switch properly pauses the priority indicated in the PFC frame.
- Verified that the end device transmits a valid PFC frame to the switch and the switch properly pauses the priority indicated in the PFC frame.

- Verified that the switch transmits a valid PFC frame to the end device and the end device properly pauses the priority indicated in the PFC frame.
- Verified that the second end device transmits a valid PFC frame to the switch and the first end device properly pauses the priority indicated in the PFC frame.

2.2.2 Test Case 2: Pairwise - ETS Test

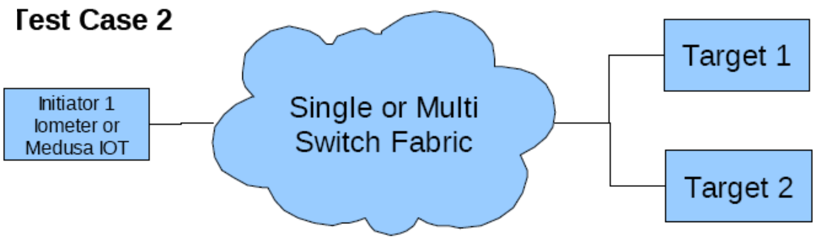


Figure 3 - ETS Test Configuration

Procedure: Link partners in this test track must support ETS as well as mapping traffic classes to VLAN priorities. The sink port is connected to a device able to count the number of frames per VLAN egressing the DUT so that the bandwidth allocations can be validated. Ideally, this device would also be able to measure latency and jitter per flow. Either PAUSE or PFC must be supported. In this test track multiple traffic class flows are used. Accepted flows and tools include:

- LAN(iperf/netperf/IxChariot/Spirent TestCenter Virtual),), SAN (IOMeter/IOZone/MedusaTools) and Finisar Xgig/Jammer.
- One of the test setups as defined in the earlier section PFC “Test Setup” can also be used.

1. ***Precursor test:*** Before enabling ETS, start each traffic class on its own.
2. ***Priority Groups:*** Configure different bandwidth percentages for at least two priorities 0, 3 and/or 6. Send at line rate traffic with priorities configured earlier 0, 3 and/or 6. Send in all combinations of the streams.
3. ***OPTIONAL* Link Strict Priority (LSP):** Enable LSP for one of the priorities. Configure different bandwidth percentages for priorities 0, 3 and/or 6. Send at line rate traffic with priorities 0, 3 and/or 6. Send in all combinations of the streams.
4. ***Jumbo Frames:*** Jumbo size the frames (including mini-jumbo 2.5kB and full jumbo 9kB, also mixing in some minimum sized frames as well) and repeat previous steps.

Observable Results:

1. Verified the amount of bandwidth that each traffic class can attain on its own.

2 Verified that you see on egress the configured frame rates (i.e. 20%, 30%, 50%) and verified that the minimum bandwidth requirements are always met.
3 *OPTIONAL*Verified when LSP traffic class is transmitting it receives majority of the link bandwidth and the other traffic classes get less than the minimum configured bandwidth if the LSP traffic class uses more.
4 Verified that the network continues to work as expected with the introduction of jumbo frames.

2.2.3 Test Track 3: Pairwise - DCBX Test

Procedure: This test verifies different states of DCBX parameters successfully negotiate or gracefully fail. This assumes that Baseline is the target DCBX version. <feature> = PFC, ETS, FCoE App

Part A:

1 Using the test setup from Test Track 1, configure the switch to distribute the configuration to the end device (and storage target, if used).
2 Transmit multiple traffic classes from the end device to the traffic generator (or use the test setup defined in Test Track 1).

Part B:

1 Use the test setup in Test Track 2, configure the switch to distribute the configuration to the end device (and storage target, if used).
2 Send line rate traffic into the ingress port from each traffic class and measure the bandwidth on the egress port (or use the test setup defined in Test Track 1).

Part C: *OPTIONAL*

Default DCBX configurations successfully negotiate to operational state

1 Unsupported DCBX features are gracefully ignored (i.e. Non-baseline matched against Baseline)
2 Disable <feature> during runtime (remote device)
3 DCBX: Stop advertising <feature> during runtime (remote device)
4 DCBX: Change <feature> configuration during runtime (remote device)

Observable Results:

Part A:

- Verified that the devices properly negotiate PFC via DCBX and the switch generates a proper PFC frame and the end device properly pauses the priority specified in the PFC frame.

Part B:

- Verified that the devices properly negotiate ETS via DCBX and the egress traffic maintains the minimum bandwidth percentage configured for each traffic class.

Part C:

- Verified different states of DCBX parameters successfully negotiate or gracefully fail

Example:

Peer A is configured: PFC: Enable=true, Willing=false, Advertise=true, Prio3 enabled

Peer B is configured: PFC: Enable=true, Willing=true, Advertise=true, Prio4 enabled

On initial DCBX negotiation, the Operational Mode of PFC should be true and the Operational configuration of Peer B will be Prio3 enabled.

If either Peer A or Peer B change the Willing variable (just one), then the Operational Mode of PFC will become 'false'.

Variation, configure Peer A and Peer B to enable the same priority (say Prio3). In this case, toggling the willing variable should have no effect on the Operational Mode of PFC.

2.2.4 Test Track 4: I/O Application Test

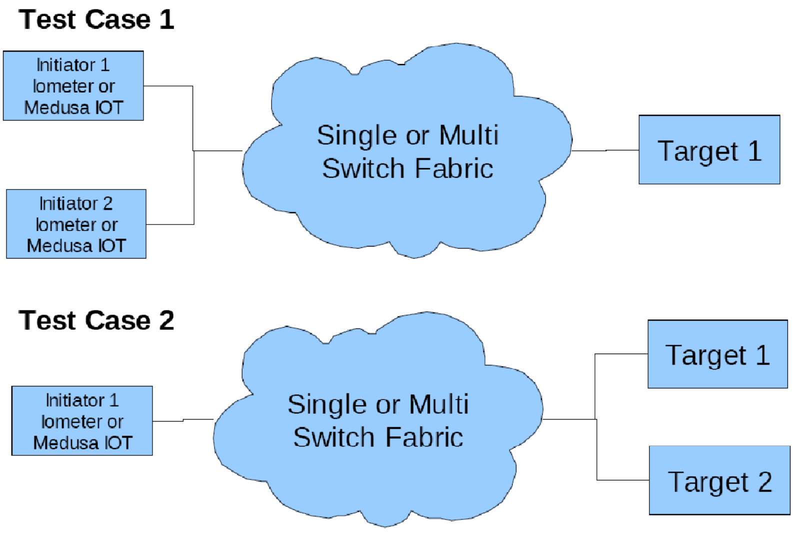

Test Case 1
Initiator 1 Iometer or Medusa IOT
Initiator 2 Iometer or Medusa IOT
Single or Multi Switch Fabric
Target 1
Test Case 2
Initiator 1 Iometer or Medusa IOT
Single or Multi Switch Fabric
Target 1
Target 2

Figure 4 - I/O Application Test Configuration

The goal of this track was to test both client/server connections through the DCB enabled switch fabric using LAN and SAN storage protocols. The configuration should entail switches interconnected with 10GbE uplinks; configure two priority queues – SAN lossless and LAN lossy. SAN priority set to use 40% of available bandwidth, LAN priority set to use 60% of available bandwidth.

End devices should at least support PFC and iSCSI or FCoE traffic generation. For LAN, use IxChariot, Spirent TestCenter Virtual), or Iperf TCP/UDP sessions on client and Server systems. For SAN, Medusa I/O test tools installed on the server side can drive line rate traffic by generating millions of IO requests. Similar to IOmeter, the tools were used to verify the network performance with real I/O applications, but more powerful on performance tests. Both these LAN and SAN tools provide the throughput statistics as well as low level link error information.

1 Use single interfaces on two initiators and one target. Using iometer settings; 3 disk volumes, 3 workers, 100% WRITE, 64KB I/Os, sequential, limit to first 8192 sectors, queue depth=32.

2 Test with SAN priority and LAN priority (for supporting protocols), test with both standard and Jumbo frames (mtu 9018), and test with READs instead of writes.

   a) ***Single Switch***: SAN WRITE test with converged LAN traffic; when running both initiators, validate that pause frames are coming back to the initiator system, and that the load is roughly equivalent between both initiators. READ test; when running both initiators validate that no pause frames are being sent to the target.

   b) ****OPTIONAL* Multiple Switches***: Tier multiple switches together. WRITE test; run the same write test as before. Compare throughput on both initiators. Should be similar to the single switch test case (with pause frames coming back to initiators). READ test; change load to read, and compare SAN and LAN throughput on both initiators. Also should be similar to the single switch case.

   c) ****OPTIONAL* Multiple Switches - different connection point***: Tier multiple switches together. Connect the initiators into different switches. WRITE test; Compare throughput and pause counts on both initiators. Should be same as single switch. READ test; Compare again, and this also should be similar to single switch.

3 Use single interfaces on two targets and one initiator. Using iometer settings; six disk volumes, six worker threads, 100% WRITE, 64KB I/Os, sequential, limit to first 8192 sectors, queue depth=32. Test with SAN priority and LAN priority (for supporting protocols), test with both standard and Jumbo frames (mtu 9018), and test with READs instead of WRITEs.

   a) ***Single Switch***: SAN WRITE test with converged LAN traffic; when running load to both targets, ensure that NO pause frames are coming back to the initiator system and/or target, and that the load is roughly equivalent between both targets. READ test; this load should cause pause frames to be sent to the targets.

   b) ****OPTIONAL* Multiple Switches***: Tier multiple switches together. WRITE test; run the same write test. Compare SAN and LAN throughput on initiator. Again, no pause frames of any type should be sent/received by any party. READ test; should have the same result as the single switch. Pause frames should be received by the targets.

c) ***OPTIONAL* Multiple Switches - different connection points***: Tier multiple switches together. Connect the targets into different switches. WRITE test; Compare throughput on both targets. No pauses should be received. READ test; pause frames should be received by the targets. Compare throughput at all points.

Observable Results:

1. Verified that during the WRITE test the Initiators were seeing pause frames and achieving roughly the same throughput and during the READ test no pause frames were being sent to the target
2. Verified that during the WRITE test no pause frames were being sent and both targets achieve roughly the same throughput and during the READ test the targets were being sent pause frames.

3 Results

Vendors were able to demonstrate interoperability, proof of concept, even at this engineering stage in the process, the ability to provide a lossless fabric simultaneously on the same network. A lot of progress was made at this event to bring about products that will interoperate in the future.

We observed that when utilizing ETS, at the same time flooding the network with traffic, the switch sent a pause frame and the traffic backed off successfully. It's important for end points, both initiators and targets, to know not to lose frames. In order to test this full bandwidth must be achieved by filling the pipe.

The screen captures presented in these results were supplied from the Finisar Xgig Jammer.

3.1 The Priority Flow Control (PFC) Test Results

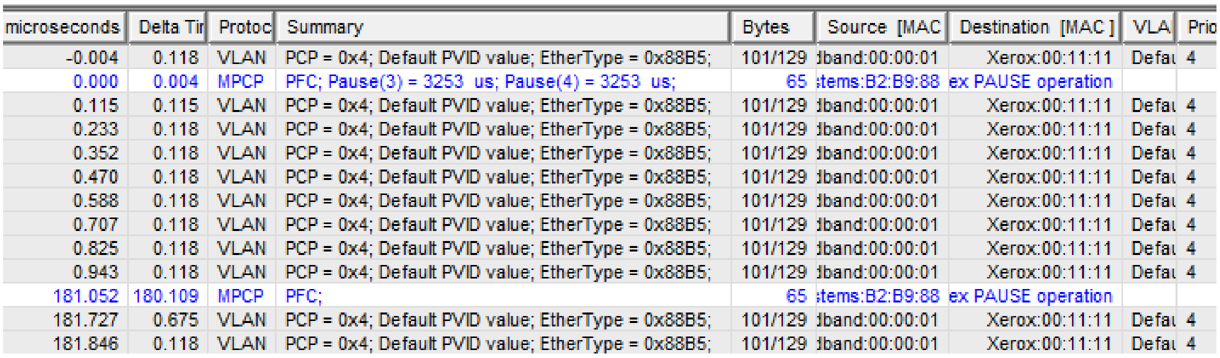
First, we tested the PFC capability of each vendor individually. We did this in two steps:

1. Prove that the vendor stops transmitting traffic when it receives a PFC Pause frame
2. Prove that the vendor generates a PFC Pause frame when it is unable to keep up with the line rate or because it has been configured to limit the bandwidth for a specific priority.

3.2 Testing the Reaction to PFC Pause Frames

To detect PFC Pause frames, we configured the Finisar Xgig protocol analyzer to capture and trigger on a PFC Pause frame. In some scenarios where the end device was not generating the Pause frames, we inserted them manually in the traffic stream using the Finisar Xgig Jammer.

Once we captured a trace with PFC pause frames. We needed to locate the PFC pause frames, identify which priorities are being paused, and filter out all the frames that do not belong to these priorities. In the picture below, the PFC frame at timestamp 0.000 is pausing priorities 3 and 4 for 3253 us each and we filtered out all frames with non-3-or-4 priorities:



microseconds	Delta Tir	Protoc	Summary	Bytes	Source [MAC	Destination [MAC]	VLA	Prio
-0.004	0.118	VLAN	PCP = 0x4; Default PVID value; EtherType = 0x88B5;	101/129	ıband:00:00:01	Xerox:00:11:11	Defaı	4
0.000	0.004	MPCP	PFC; Pause(3) = 3253 us; Pause(4) = 3253 us;	65	:tems:B2:B9:88	ex PAUSE operation		
0.115	0.115	VLAN	PCP = 0x4; Default PVID value; EtherType = 0x88B5;	101/129	ıband:00:00:01	Xerox:00:11:11	Defaı	4
0.233	0.118	VLAN	PCP = 0x4; Default PVID value; EtherType = 0x88B5;	101/129	ıband:00:00:01	Xerox:00:11:11	Defaı	4
0.352	0.118	VLAN	PCP = 0x4; Default PVID value; EtherType = 0x88B5;	101/129	ıband:00:00:01	Xerox:00:11:11	Defaı	4
0.470	0.118	VLAN	PCP = 0x4; Default PVID value; EtherType = 0x88B5;	101/129	ıband:00:00:01	Xerox:00:11:11	Defaı	4
0.588	0.118	VLAN	PCP = 0x4; Default PVID value; EtherType = 0x88B5;	101/129	ıband:00:00:01	Xerox:00:11:11	Defaı	4
0.707	0.118	VLAN	PCP = 0x4; Default PVID value; EtherType = 0x88B5;	101/129	ıband:00:00:01	Xerox:00:11:11	Defaı	4
0.825	0.118	VLAN	PCP = 0x4; Default PVID value; EtherType = 0x88B5;	101/129	ıband:00:00:01	Xerox:00:11:11	Defaı	4
0.943	0.118	VLAN	PCP = 0x4; Default PVID value; EtherType = 0x88B5;	101/129	ıband:00:00:01	Xerox:00:11:11	Defaı	4
181.052	180.109	MPCP	PFC;	65	:tems:B2:B9:88	ex PAUSE operation		
181.727	0.675	VLAN	PCP = 0x4; Default PVID value; EtherType = 0x88B5;	101/129	ıband:00:00:01	Xerox:00:11:11	Defaı	4
181.846	0.118	VLAN	PCP = 0x4; Default PVID value; EtherType = 0x88B5;	101/129	ıband:00:00:01	Xerox:00:11:11	Defaı	4

Figure 5 - PFC Test Results

The 2nd PFC frame at timestamp 181.052 is a Pause release since it contained 0-valued Pause times for all priorities. Note that there were still 8 frames sent after the Pause frame was issued, the last frame being sent 0.943 microseconds after the Pause. These are in-flight frames. According to the specification, a transmitter can still send frames for up to 1.4 µs after a Pause frame was received. After these 8 frames, the transmitter stopped sending for 180 microseconds, until it received the Pause Release frame, and then it resumed. This proves that this device reacts to Pause frames adequately.

After verifying manually the proper reaction to PFC Pause frames a couple times, we configured the analyzer to do it automatically on all captures. This way, we proved that each vendor device was actually pausing when requested to, but they take more or less time to stop transmitting frames after receiving the pause. Some vendors even take longer than allowed 1.4us in the specification.

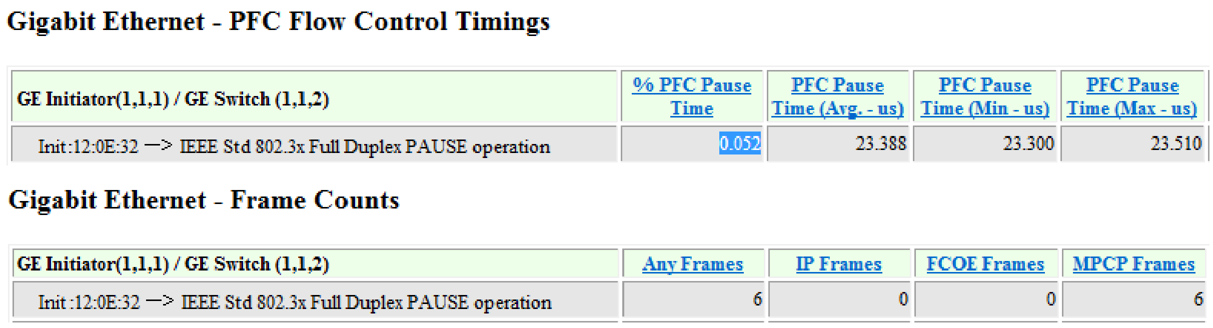
3.3 Testing the Generation of PFC Pause Frames

Generation of PFC may be tested with:

- iSCSI and FCoE Initiator/Target devices
- Switches

iSCSI Initiator/Targets do not automatically generate PFC Pause frames since they usually rely on the TCP stack ability to recover frame losses. Fibre Channel relies on a lossless fabric.

In one scenario during testing we performed 750 MB/sec Read operations from a storage initiator through the switch, to a storage target. At that rate, the initiator transmitted PFC Pause frames to the switch. The network analyzer software reported six (6) Pause frames in the capture, for an average pause time of 23.388 µs, the overall traffic being paused 0.052% of the time:



Gigabit Ethernet - PFC Flow Control Timings

GE Initiator(1,1,1) / GE Switch (1,1,2)	% PFC Pause Time	PFC Pause Time (Avg. - us)	PFC Pause Time (Min - us)	PFC Pause Time (Max - us)
Init:12:0E:32 —> IEEE Std 802.3x Full Duplex PAUSE operation	0.052	23.388	23.300	23.510

Gigabit Ethernet - Frame Counts

GE Initiator(1,1,1) / GE Switch (1,1,2)	Any Frames	IP Frames	FCOE Frames	MPCP Frames
Init :12:0E:32 —> IEEE Std 802.3x Full Duplex PAUSE operation	6	0	0	6

Figure 6 - PFC Flow Control Results

As opposed to Initiator/Target devices, it was easier to test a Switch's ability to generate PFC Pause frames since we could configure them to limit the storage traffic to a certain bandwidth using the ETS features.

3.4 The ETS Test Results

The ETS consists of allocating bandwidth per traffic priority group. This requires each frame to contain a VLAN header with the 3-bit priority field (called PCP) set to a proper value.

In ETS, there are 8 possible traffic priorities, 0 through 7, and a minimum bandwidth as percentage of full line rate, is allocated to each priority. ETS utilizes the concept of Priority Group to group multiple priorities together, but we will assume only 1 priority per group for following discussion.

In one scenario, we configured the ETS feature of a switch as follows:

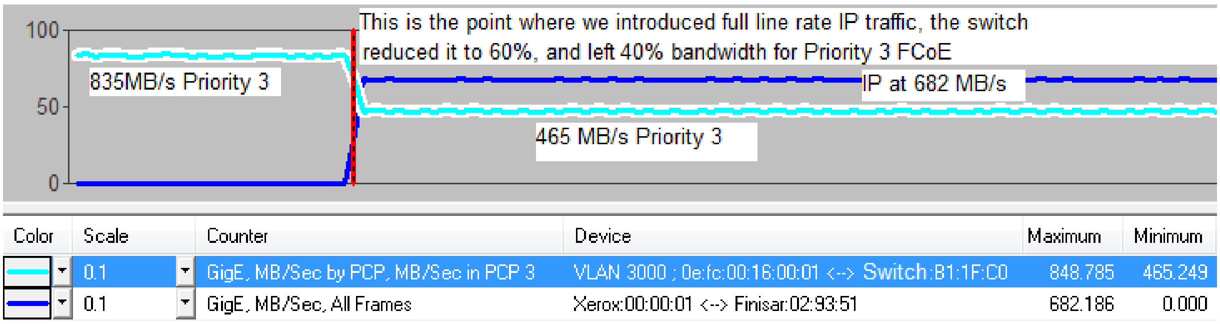
- Priority 3: Minimum of 40% full line rate bandwidth
- Other Priorities: Minimum of 60% full line rate bandwidth

With 10 Gbps traffic, this translates to:

- Priority 3: Minimum of 480 MB/s bandwidth
- Other Priorities: Minimum of 720 MB/s bandwidth

Then we performed 835 MB/s Read operations on Priority 3 from a storage initiator, through the switch to a storage target. At this point, there were no PFC Pause frames on the wire, the traffic was flowing freely from the initiator to the target even though it was well over the minimum 480 MB/s. That was because there was no non-Priority 3 traffic on the link yet.

For the ETS feature to become active, we needed to flood the link between the initiator and the Switch with non-Priority 3 traffic. We used a load tester to send full line rate IP traffic through the switch to the initiator. At that point, the switch started to enforce the ETS feature. It dropped frames on the IP traffic and it sent PFC Pause Frames to the Priority 3 FCoE target. The load tester's software exhibits the ETS behavior as shown below. The light blue line is the Priority 3 traffic, whereas the dark blue line is the IP traffic. The red line shows where the IP traffic started and where the switch forced the Priority 3 traffic to reduce to 465 MB/s, or roughly 40% of the full line rate:



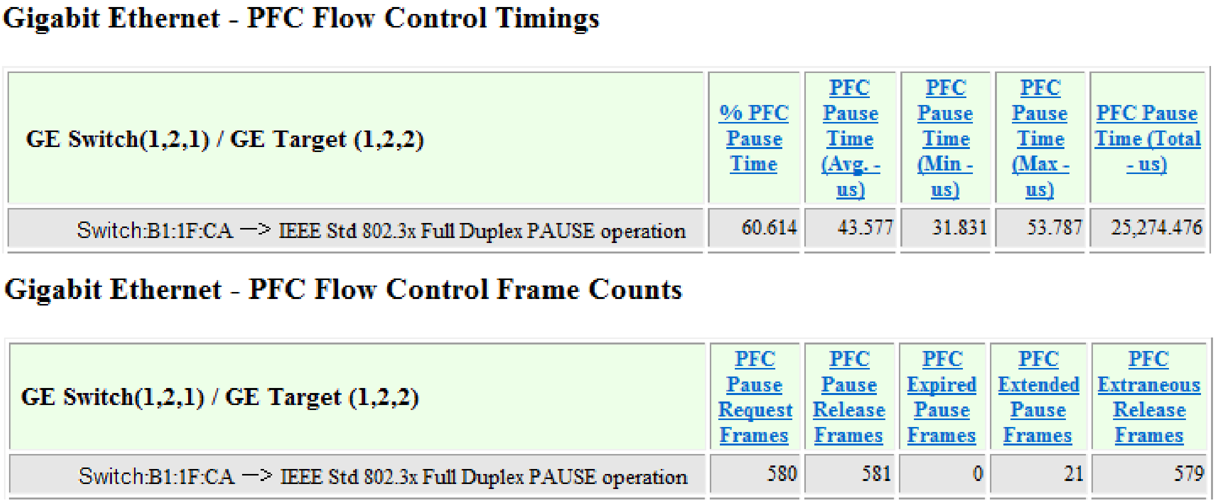
This is the point where we introduced full line rate IP traffic, the switch reduced it to 60%, and left 40% bandwidth for Priority 3 FCoE

Color	Scale	Counter	Device	Maximum	Minimum
—	0.1	GigE, MB/Sec by PCP, MB/Sec in PCP 3	VLAN 3000 ; 0e:fc:00:16:00:01 <--> Switch:B1:1F:C0	848.785	465.249
—	0.1	GigE, MB/Sec, All Frames	Xerox:00:00:01 <--> Finisar:02:93:51	682.186	0.000

Figure 7 - ETS Test Results

To ensure that no frames were dropped on the FCoE target side, we enabled cross-port analysis. In this mode, the tester ensures that all the frames on one side of the switch are also found on the other side and they are in the same order. The tester didn't report frame losses on the FCoE target side. This means that the Switch successfully flow controlled the traffic.

By generating a report on the Switch - FCoE target side, we can see that the link spent 60.614% of the entire time in Pause state. That was achieved through 580 PFC Pause Frames sent on Priority 3 by the Switch for an average pause time of 43.577 us.



Gigabit Ethernet - PFC Flow Control Timings

GE Switch(1,2,1) / GE Target (1,2,2)	% PFC Pause Time	PFC Pause Time (Avg. - us)	PFC Pause Time (Min - us)	PFC Pause Time (Max - us)	PFC Pause Time (Total - us)
Switch:B1:1F:CA —> IEEE Std 802.3x Full Duplex PAUSE operation	60.614	43.577	31.831	53.787	25,274.476

Gigabit Ethernet - PFC Flow Control Frame Counts

GE Switch(1,2,1) / GE Target (1,2,2)	PFC Pause Request Frames	PFC Pause Release Frames	PFC Expired Pause Frames	PFC Extended Pause Frames	PFC Extraneous Release Frames
Switch:B1:1F:CA —> IEEE Std 802.3x Full Duplex PAUSE operation	580	581	0	21	579

Figure 8 - ETS Pause Results

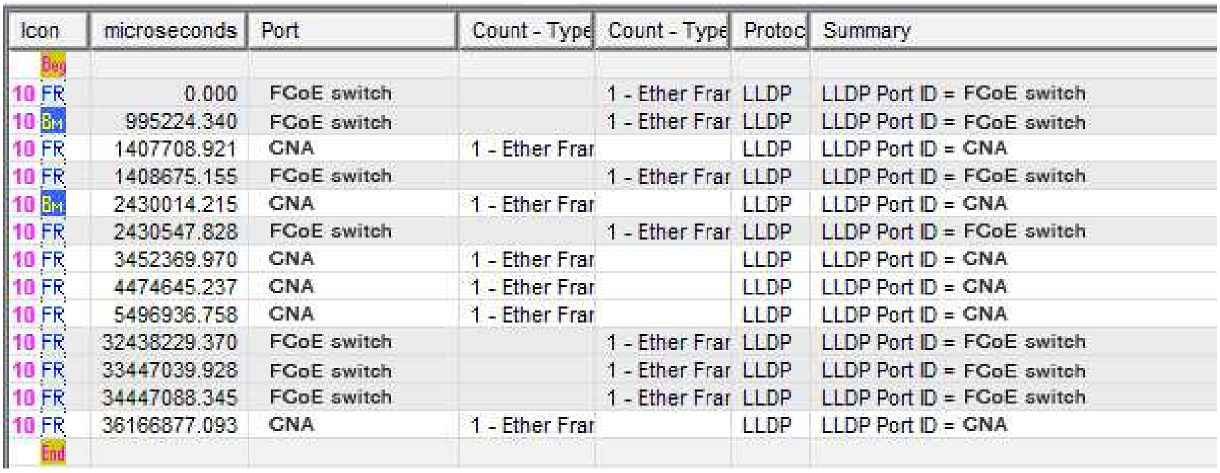
In an unexpected manner, the percent Pause Time metric turned out to be extremely useful to test the ETS feature. We had configured the FCoE traffic on Priority 3 to 40% bandwidth at full line rate, and here we show that the FCoE traffic was paused 60% of the time on the other side of the switch, leaving the traffic through 40% of the time at full line rate. This metric proves in itself that the ETS feature worked in this case.

Overall, we tested that the switch's ability to generate PFC Pause frames at the same time as we tested the ETS feature. This way we proved that the switches were enforcing ETS and they were generating PFC Pause frames when needed.

3.5 The DCBX Test Results

The DCB Capability eXchange Protocol (DCBX) adds new TLVs in an LLDP frame. It is used to announce ETS and PFC configurations on a link-by-link basis. Each switch and end device announces its ETS and PFC configurations through DCBX messages to its peers. Each peer has a choice of accepting or ignoring that configuration. The DCBX announcements occur several times when an end device connects to a switch, and then every 30 seconds or so. The DCBX protocol defines how link capabilities are discovered and what happens in the case of conflicting configurations on a per feature basis.

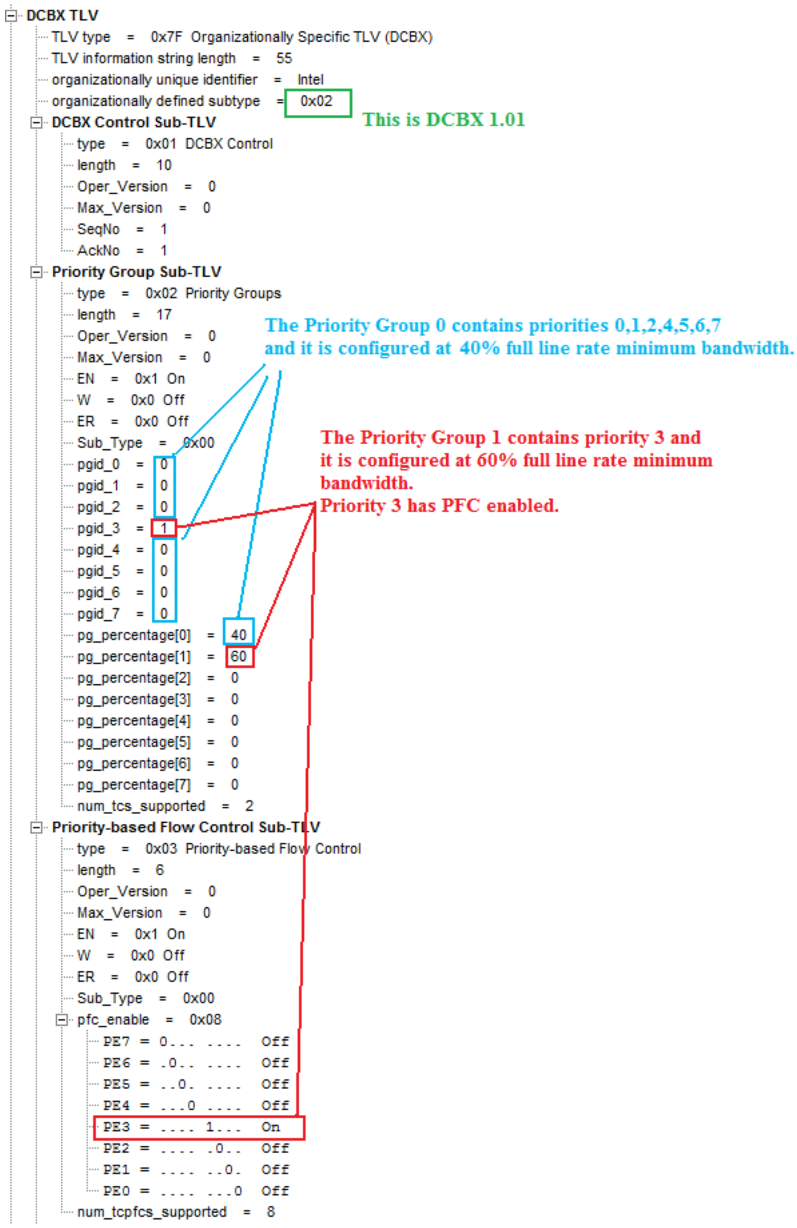
To test DCBX, we connected end devices and switches and we captured the DCBX exchanges. The capture below shows some DCBX exchanges between the CNA end device and the switch.



Icon	microseconds	Port	Count - Type	Count - Type	Protoc	Summary
Beg						
10 FR	0.000	FCoE switch		1 - Ether Frar	LLDP	LLDP Port ID = FCoE switch
10 BM	995224.340	FCoE switch		1 - Ether Frar	LLDP	LLDP Port ID = FCoE switch
10 FR	1407708.921	CNA	1 - Ether Frar		LLDP	LLDP Port ID = CNA
10 FR	1408675.155	FCoE switch		1 - Ether Frar	LLDP	LLDP Port ID = FCoE switch
10 BM	2430014.215	CNA	1 - Ether Frar		LLDP	LLDP Port ID = CNA
10 FR	2430547.828	FCoE switch		1 - Ether Frar	LLDP	LLDP Port ID = FCoE switch
10 FR	3452369.970	CNA	1 - Ether Frar		LLDP	LLDP Port ID = CNA
10 FR	4474645.237	CNA	1 - Ether Frar		LLDP	LLDP Port ID = CNA
10 FR	5496936.758	CNA	1 - Ether Frar		LLDP	LLDP Port ID = CNA
10 FR	32438229.370	FCoE switch		1 - Ether Frar	LLDP	LLDP Port ID = FCoE switch
10 FR	33447039.928	FCoE switch		1 - Ether Frar	LLDP	LLDP Port ID = FCoE switch
10 FR	34447088.345	FCoE switch		1 - Ether Frar	LLDP	LLDP Port ID = FCoE switch
10 FR	36166877.093	CNA	1 - Ether Frar		LLDP	LLDP Port ID = CNA
End						

Figure 9 - DCBX Test Results

The DCBX section of the LLDP frame contains configuration information about ETS (i.e. bandwidth allocation per Priority Group) and PFC among others. Following is a portion of the DCBX section in one of the frames:



```
DCBX TLV
  TLV type  =  0x7F  Organizationally Specific TLV (DCBX)
  TLV information string length  =  55
  organizationally unique identifier  =  Intel
  organizationally defined subtype  =  0x02
  DCBX Control Sub-TLV
    type  =  0x01  DCBX Control
    length  =  10
    Oper_Version  =  0
    Max_Version  =  0
    SeqNo  =  1
    AckNo  =  1
  Priority Group Sub-TLV
    type  =  0x02  Priority Groups
    length  =  17
    Oper_Version  =  0
    Max_Version  =  0
    EN  =  0x1  On
    W  =  0x0  Off
    ER  =  0x0  Off
    Sub_Type  =  0x00
    pgid_0  =  0
    pgid_1  =  0
    pgid_2  =  0
    pgid_3  =  1
    pgid_4  =  0
    pgid_5  =  0
    pgid_6  =  0
    pgid_7  =  0
    pg_percentage[0]  =  40
    pg_percentage[1]  =  60
    pg_percentage[2]  =  0
    pg_percentage[3]  =  0
    pg_percentage[4]  =  0
    pg_percentage[5]  =  0
    pg_percentage[6]  =  0
    pg_percentage[7]  =  0
    num_tcs_supported  =  2
  Priority-based Flow Control Sub-TLV
    type  =  0x03  Priority-based Flow Control
    length  =  6
    Oper_Version  =  0
    Max_Version  =  0
    EN  =  0x1  On
    W  =  0x0  Off
    ER  =  0x0  Off
    Sub_Type  =  0x00
    pfc_enable  =  0x08
      PE7 = 0... ....  Off
      PE6 = .0.. ....  Off
      PE5 = ..0. ....  Off
      PE4 = ...0 ....  Off
      PE3 = .... 1...  On
      PE2 = .... .0..  Off
      PE1 = .... ..0.  Off
      PE0 = .... ...0  Off
    num_tcpfcs_supported  =  8
```

This is DCBX 1.01

The Priority Group 0 contains priorities 0,1,2,4,5,6,7 and it is configured at 40% full line rate minimum bandwidth.

The Priority Group 1 contains priority 3 and it is configured at 60% full line rate minimum bandwidth.
Priority 3 has PFC enabled.

Figure 10 - Configuring Priorities

In the DCBX frame above we can see that:

- Priority 3 is assured a minimum of 60% of full bandwidth (720 MB/s) and it has PFC enabled
- The other priorities have a minimum of 40% of full bandwidth (480 MB/s) and they have PFC disabled.

Each feature in the DCBX frame as a Willing (W) bit associated. The Willing bit is usually set to 0 bit the switch, and 1 by the end device. A Willing bit of 1 means that the device is willing to accept the configuration of the switch. We saw scenarios where the switch proposed an ETS configuration (Priority Groups) of 60% bandwidth on Priority 3, 40% for other Priorities, but the end device was proposing a ratio of 50/50.

3.6 The Multi-Vendor Test Results

For one of the final tests, multiple initiators and targets were connected to a switch and various tests for ETS and PFC were performed simultaneously. The test methodology is described below.

One of these configurations had two initiators and two targets connected through a switch. One initiator was writing to both the FCoE target and the iSCSI target. The other initiator was writing to the iSCSI target only. We had the switch configured to limit traffic for Priority 3 (FCoE traffic) at 10%, limit, and 90% for other Priorities. Using a strict 10% priority means that the line rate is limited to 10% bandwidth, i.e. 120 MB/s even if there is no additional traffic. The following picture shows the throughputs for each type of traffic on each link after running for some time.

On the FCoE side, the switch was pausing Initiator1 at 99 MB/s traffic on Priority 3. It was only pausing .1% of the time. This is probably because Initiator1 reached the 120 MB/s rate (corresponds to 10%) in short bursts and the switch reacted to these short bursts with short Pause frames, overall pausing only .1% of the entire time.

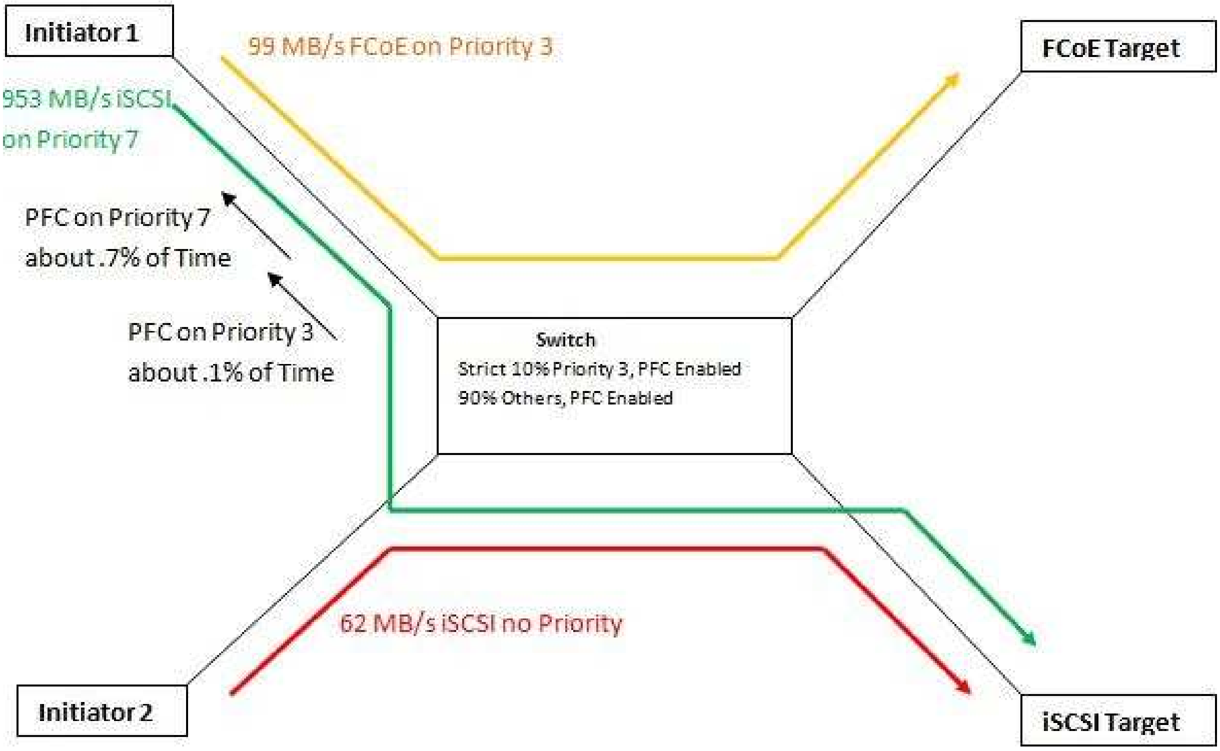

Initiator 1
99 MB/s FCoE on Priority 3
FCoE Target
953 MB/s iSCSI on Priority 7
PFC on Priority 7 about .7% of Time
PFC on Priority 3 about .1% of Time
Switch
Strict 10% Priority 3, PFC Enabled
90% Others, PFC Enabled
62 MB/s iSCSI no Priority
Initiator 2
iSCSI Target

Figure 11 - Multi-vendor Test

The same happened on the iSCSI target side. The switch was configured to 90% for all Priorities except 3, so the traffic on Priority 7 was paused sometimes when reaching 953 MB/s. Again that traffic was probably reaching the 1080 MB/s threshold (which corresponds to 90% line rate) in short bursts, so the switch ended up pausing it .7% of the time.

Overall, the switch behaved properly in this configuration.

4 Conclusions

Participants at this plugfest stated that such interoperability events are essential at this early stage of standards development. DCBX

Although the expectations were low with respect to performance testing at this first plugfest, the results were quite impressive. All the participating vendors were able to demonstrate interoperable solutions for DCB components and end-to-end solutions that used "lossless" DCB environment.

5 About the Ethernet Alliance

The Ethernet Alliance is a consortium that includes system and component vendors, industry experts and university and government professionals who are committed to the continued success and expansion of Ethernet technology. The Ethernet Alliance takes IEEE 802 Ethernet standards to market by supporting activities that span from incubation of new Ethernet technologies to interoperability demonstrations, certification and education.

A link to the press release for this event can be found at http://tinyurl.com/m3x8jf. For more information, visit www.ethernetalliance.org.

6 Glossary of Terms

Data Center Bridging capability eXchange Protocol (DCBX): A protocol that is used for conveying capabilities and configuration of DCB features between link neighbors to ensure consistent configuration across the network. This protocol is expected to leverage functionality provided by 802.1AB (LLDP)...

Enhanced Transmission Selection (ETS): IEEE 802.1Qaz task force is defining Enhanced Transmission Selection (ETS) that provides a common management framework for assignment of bandwidth to traffic classes.

Fibre Channel Over Ethernet (FCoE): The proposed mapping of Fibre Channel frames over full duplex IEEE 802.3 networks.

Internet Small Computer System Interface (iSCSI): An Internet Protocol (IP)-based storage networking standard for linking data storage facilities. By carrying SCSI commands over IP networks, iSCSI is used to facilitate data transfers over intranets and to manage storage over long distances.

Link Layer Discovery Protocol (LLDP): A vendor-neutral Layer 2 protocol that allows a network device to advertise its identity and capabilities on the local network. The protocol is formally defined as IEEE standard 802.1AB-2005 ..

Priority-based Flow Control (PFC): IEEE 802.1Qbb is standardizing Priority-based Flow Control (PFC), which provides a link level flow control mechanism that can be controlled independently for each priority. The goal of this mechanism is to ensure zero loss due to congestion in DCB networks.

Type-Length-Value (TLV): Within data communication protocols, optional information may be encoded as a **type-length-value** or **TLV** element inside of the protocol.

The type and length fields are fixed in size (typically 1-4 bytes), and the value field is of variable size. These fields are used as follows:

Type: A numeric code which indicates the kind of field that this part of the message represents.

Length: The size of the value field (typically in bytes).

Value: Variable sized set of bytes which contains data for this part of the message.