Priority Flow Control: Build Reliable Layer 2 Infrastructure

What You Will Learn

This document describes priority flow control (PFC), one of the functions of IEEE 802.1 Data Center Bridging, introduced to address reliability at Layer 2 on the Ethernet medium. PFC enables lossless semantics for a subset of the Layer 2 flows carried on an Ethernet segment. The subset of flows subject to the lossless semantics is identified by the IEEE 802.1p class of service (CoS) information found in the IEEE 802.1Q tags. PFC is currently being defined in the IEEE 802.1Qbb workgroup. Although the Cisco® proposal for PFC has not achieved standards status yet, a significant number of vendors have embraced the frame format proposed by Cisco, and therefore PFC implementations can be deployed today without risk of future incompatibility in that respect. PFC is an important requirement for enabling Fibre Channel over Ethernet (FCoE) for storage consolidation, and this document describes how the Cisco Nexus™ 5000 Series Switches use PFC to build a lossless Layer 2 medium for FCoE.

What Is Priority Flow Control?

Traditional IEEE 802.3 Ethernet defines an unreliable communication medium; it does not offer guarantees that a packet injected into the network will arrive at its intended destination. Reliability is expected by means of upper-layer protocols and is outside the scope of the initial definition.

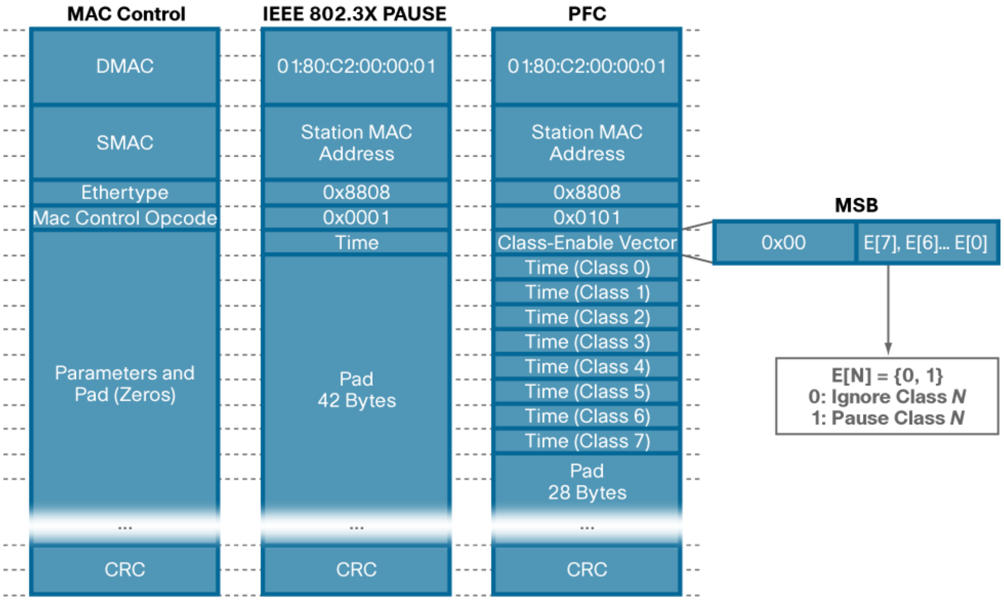
In a network path that normally consists of multiple hops between source and destination, lack of feedback between transmitters and receivers at each hop is one of the main causes of unreliability. Transmitters can send packets faster than receivers accept packets, and as the receivers run out of available buffer space to absorb incoming flows, they are forced to silently drop all traffic that exceeds their capacity. These semantics work fine at Layer 2, so long as upper-layer protocols handle drop-detection and retransmission logic.

For applications that cannot build reliability on upper layers, the addition of flow control functions at Layer 2 can offer a solution. Flow control enables feedback from a receiver to its sender to communicate buffer availability. Its first implementation in IEEE 802.3 Ethernet uses the IEEE 802.3x PAUSE control frames. IEEE 802.3x PAUSE is defined in Annex 31B of the IEEE 802.3 specification. Simply put, a receiver can generate a MAC control frame and send a PAUSE request to a sender when it predicts the potential for buffer overflow. Upon receiving a PAUSE frame, the sender responds by stopping transmission of any new packets until the receiver is ready to accept them again.

IEEE 802.3x PAUSE works as designed, but it suffers a basic disadvantage that limits its field of applicability: after a link is paused, a sender cannot generate any more packets. As obvious as that seems, the consequence is that the application of IEEE 802.3x PAUSE makes an Ethernet segment unsuitable for carrying multiple traffic flows that might require different quality of service (QoS). Thus, enabling IEEE 802.3x PAUSE for one application can affect the performance of other network applications.

IEEE 802.1Qbb PFC extends the basic IEEE 802.3x PAUSE semantics to multiple CoSs, enabling applications that require flow control to coexist on the same wire with applications that perform better without it. PFC uses the IEEE 802.1p CoS values in the IEEE 802.1Q VLAN tag to differentiate up to eight CoSs that can be subject to flow control independently. The differences between IEEE 802.3x PAUSE and PFC frames are shown in Figure 1.

Figure 1. IEEE 802.3x PAUSE and PFC Frame Format



As Figure 1 shows, a 64-byte MAC control frame is used by both IEEE 802.3x PAUSE and PFC. In both cases, numeric values can be used to describe the requested duration of PAUSE. However, since PFC acts independently on eight different CoSs, the frame describes the PAUSE duration for each CoS.

The PAUSE duration for each CoS is a 2-byte value that expresses time as a number of quanta, where each represents the time needed to transmit 512 bits at the current network speed. A PAUSE duration of zero quanta has the special meaning of unpausing a CoS. Typical implementations will not try to guess a specific duration for PAUSE, instead relying on the X-ON and X-OFF style behavior that can be obtained by setting PAUSE for a large number of quanta and then explicitly resuming traffic when appropriate..

Distance Limitations

A receiver using PFC must predict the potential for buffer exhaustion for a CoS, and respond by generating an explicit PAUSE frame for that CoS when that condition arises. The PAUSE frame needs to be sent back to the other end of the wire early enough, so that the talkative sender has time to stop transmitting before buffers overflow on the receiving side.

Obviously, since bits on a wire travel at a finite speed, the length of the wire affects how early the receiving end must act. The longer the wire, the earlier a receiver must send back a PAUSE frame.

Put another way, at any point in time the receiver must have enough residual buffers available to store any packet that might be in flight while the PAUSE frame travels back to the sender and gets processed there. Since after the PAUSE request has been sent, the system “transmitter + wire + receiver” must drain all existing packets into receiver buffers, the definition of an appropriate buffer threshold on the receiver side is critical to a functioning PFC implementation.

Definition of Receiver Buffer Threshold

Consider the case of a stream of packets flowing from sender S to receiver R through cable C. To set up the receiver R thresholds for an appropriate PAUSE implementation, the following factors need to be considered:

- **Maximum transmission unit (MTU) on the transmitting end of receiver R:** If R intends to emit a PAUSE frame, the PAUSE frame cannot preempt another frame currently being transmitted in the same direction (from R to S). You can safely assume that the PAUSE frame will pass ahead of any other packet queued in R for transmission, but you do not want to corrupt a packet that is currently in flight. In the worst case, R generates a PAUSE frame right when the first bit of a maximum-size packet MTU[R] has started engaging the transmission logic. The maximum-size packet in flight in front of the PAUSE frame delays the emission of the PAUSE frame; as such, it can belong to any CoS, and we'll need to account for the largest MTU configured across all CoSs.
- **Speed of wire W:** Bits travel at 70 percent of the speed of light in a twin-ax copper cable, and at 65 percent of the speed of light in a single-mode fiber. Therefore, every 100 meters of cable delays reception of a packet by an additional 476 nanosecond (ns) for copper cables and by an additional 513 ns for single-mode fibers. At 10 Gbps, while a PAUSE frame spends 476 ns traversing 100 meters of copper cable, sender S will transmit an additional 4760 bits (595 bytes) in the opposite direction, or an additional 5130 bits (641 bytes) for the 513 ns of delay in a single-mode fiber. The rest of this discussion will take into account the worst case: 641 bytes for every 100 meters.

  If you view the cable as a buffer in itself, this buffer holds a certain amount of data when the PAUSE frame starts its journey. The amount of data "stored" in the cable when the PAUSE frame is injected is exactly the same as the amount of new data that is injected by sender S while the PAUSE frame traverses the wire. Therefore, combining the previous element with this one, overall you need to double the contribution to the threshold that every 100 meters of cable adds. That brings the total cable length contribution to 1282 bytes for every 100 meters, or approximately 1300 bytes. This is the only variable element in the definition of the receiver thresholds.
- **Transceiver latency:** As with the speed of the wire, the latency introduced by both the transceivers in R (the transmitting end of the PAUSE frame) and S (the receiving end of the PAUSE frame) delays the PAUSE frame and therefore maps directly to a certain amount of additional data generated in the opposite direction from S to R. Again, every 512 ns of latency mean an additional 640 bytes to account for, and again, the contribution must be doubled, yielding 1280 bytes per 512 ns of combined transceivers latency. In general, though, the contribution of Small Form-Factor Pluggable Plus (SFP+) transceivers to the overall byte count is negligible (less than 512 ns) compared to all the other elements listed here, and therefore this piece can be ignored moving forward[1].
- **Response time of sender S:** After a PAUSE frame has been received by S, the internal logic of S will spend an implementation-dependent amount of time processing the frame and responding to the request to stop transmission. The PFC definition caps this time to 60 quanta for any implementation. As seen previously, 60 quanta are 512 * 60 = 30,720 bits (3840 bytes).
- **MTU on the transmitting end of sender S:** After sender S finally decides to comply with the PAUSE request made by receiver R, S can stop only at packet boundaries, to avoid packet corruptions. In the worst case, S will have completed processing of the PAUSE frame when the first bit of a maximum-size packet MTU[S] has started engaging the transmission logic. Contrary to MTU[R], MTU[S] impacts the receiver's buffers only if it's meant for the CoS that triggered the PAUSE in the first place, since packets destined to other CoS will use a different buffer pool; as such, the worst case for MTU[S] is the MTU configured for that CoS.

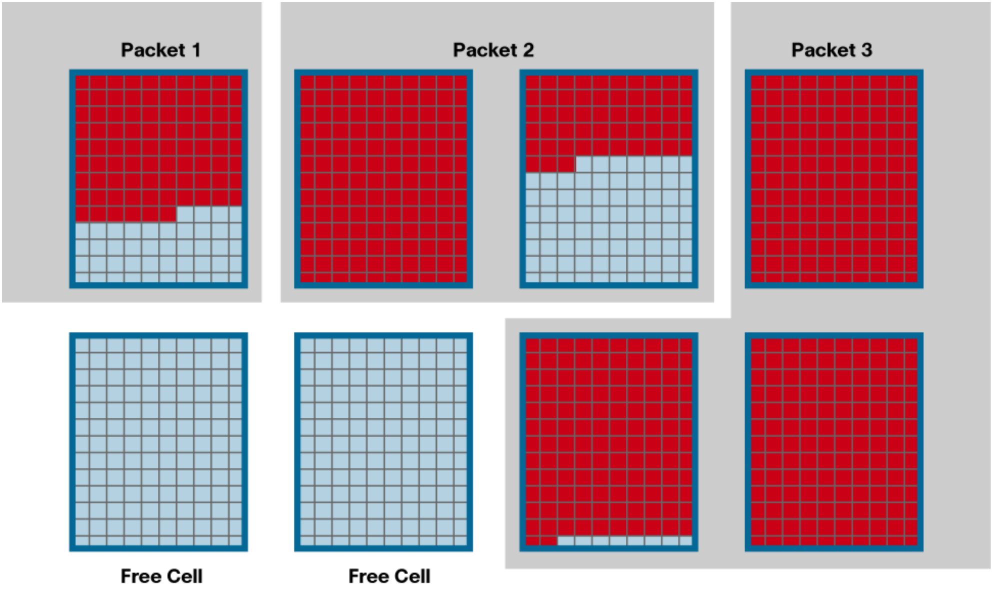
[1] This assumption can be made because none of the implementations at Cisco currently supports 10GBASE-T transceivers. With 2.5 microseconds of latency per transceiver, the current state of the art 10GBASE-T transceivers would account for the equivalent of a total of 12,800 bytes of delay, and it would not be possible to ignore such a large amount of additional buffering

If we combine all these elements, we end up with receiver thresholds that need to account for a fixed overhead of MTU[R] + MTU[S] + 3,840 bytes, and a variable amount of buffering that is a function of the cable length: approximately 1300 bytes for every 100 meters of cable. Later in this document, you will see how these factors apply in one exemplary implementation, the Cisco Nexus 5000 Series Switch.

Buffers and Packets

The math just discussed ignores a very important point. The math describes the number of bytes emitted by a sender after a receiver has concluded that it is time to generate a PAUSE frame. The bytes are broadcast as a contiguous flow of packets, but when they are received, they cannot be buffered as a contiguous flow. The internal buffer management strategy used by a receiver normally allocates one or more noncontiguous blocks of memory for storing each packet. Every implementation is different, but general considerations about block allocation efficiency, management overhead, and buffer fragmentation determine the optimal block size for an implementation (Figure 2).

Figure 2. Buffer Memory and Cells



For example, the Cisco Nexus 5000 Series' unified port controller (UPC) subdivides the 480 KB of port buffers into 160-byte units called cells. Packets buffered by a receiver consume memory in cell multiples, regardless of their actual size. A 64-byte packet will consume one cell of 160 bytes, and 96 bytes of this cell will remain unused while the packet sits in the UPC's buffers (typically, a cell cannot be shared by multiple packets). Ninety-six unused bytes for a 64-byte packet represent a 60 percent overhead for that packet size. Obviously, 64-byte packets produce the maximum overhead for this implementation.

You must consider the minimum unit of buffer allocation for a packet, because it affects the way that the contiguous flow of bytes from the sender consumes receiver buffers. See the next section for information about how to modify the Cisco Nexus 5000 Series' threshold to account for this factor.

The transition from contiguous bytes on the wire to quantized chunks is highly implementation dependent: the Cisco Nexus 5000 Series Switches have 160-byte cells with the worst case at 64-byte packets, the Cisco Nexus 2000 Series Fabric Extenders have 80-byte cells with the worst case at 81-byte packets, and the Menlo ASIC on the Emulex or QLogic Converged Network Adapters (CNAs) uses yet another buffer allocation strategy. Although the math for the contiguous stream remains the same, the actual choice of thresholds varies depending on this additional factor.

A Real-Life Example: FCoE on the Cisco Nexus 5000 Series

The Cisco Nexus 5000 Series uses PFC to establish a lossless medium for the Fibre Channel payload in the FCoE implementation. As of Cisco NX-OS Software Release 4.0(1a)N1(1) for the Cisco Nexus 5000 Series, the MTU for the CoS dedicated to FCoE is set to 2240 bytes, so that's the value we want to use for MTU[S], while MTU[R] must be set to 9216 bytes (Ethernet jumbo frame). Furthermore, the receiver threshold for the FCoE CoS is computed based on an assumed maximum of 300 meters of cable between sender and receiver. This data added to the calculations described earlier yield a total of 2240 + 9216 + 3840 + 3 * 1300 bytes = 19,196 bytes. That is, the receiver logic on the UPC in the Cisco Nexus 5000 Series needs to generate a PFC PAUSE request for FCoE when its CoS buffers fill to the point that the available free buffers can accommodate only 19,196 bytes of additional data packets.

As discussed in the previous section, these 19,196 bytes of data packets need to be stored in buffer cells, and for the Cisco Nexus 5000 Series the worst case arises when each packet in the stream is 64 bytes in size. These 19,196 bytes map to 300 minimum-size 64-byte packets, and therefore those 300 packets will actually consume 300 cells, for a total of 48,000 bytes; the software running on the Cisco Nexus 5000 Series sets the receiver threshold in such a way that when only 300 cells are left for the FCoE CoS, a PAUSE is sent back to the sender.

Pay attention to the fact that the math above assumes an FCoE CoS with an MTU of 2240 bytes. Adding a lossless CoS for regular Ethernet traffic with an MTU[S] of 9216 bytes would obviously cost more buffers. Specifically, you would need to absorb 2 * 9216 + 3840 + 3 * 1300 bytes = 26,172 bytes, and therefore you would need to put the receiver threshold at no less than 409 cells for that class (65,440 bytes of UPC buffer memory).

In general, the current software on the Cisco Nexus 5000 Series can support up to three lossless CoSs: one for FCoE (MTU at 2240 bytes, nonconfigurable), one with a jumbo MTU of 9216 bytes, and one with the traditional Ethernet MTU of 1500 bytes.

Number of Cells for PFC over a 10-Kilometer Link

Any extension of PFC above the current nonconfigurable restriction of 300 meters maximum distance would imply a different configuration for the receiver threshold. For FCoE with the same MTU[S] of 2240 bytes, for example, a 10-kilometer link would put the receiver threshold at a value that corresponds to packet data equivalent to 2240 + 9216 + 3840 + 100 * 1300 bytes = 145,296 bytes. A 64-byte packet size worst case would mean 2271 packets: that is, 2271 cells of 160 bytes. This option would therefore put the FCoE receiver threshold at 363,360 bytes, meaning that when 355 KB of buffers remain available for the FCoE CoS, a PAUSE will be sent back. Link utilization considerations (see the next section) require that at least a few packets flow on the 10-kilometer link before sending the PAUSE frame. These buffers aiding link utilization would be counted on top of the 355 KB needed just for the lossless semantics on the FCoE CoS. Given that a total of 480 KB of buffers is available on any Cisco Nexus 5000 Series interface, you can see how little room this would leave for other CoSs sharing the same 10-kilometer link. Cisco has no plans to extend PFC to such distances at the current time.

Link Utilization

Defining the appropriate receiver buffer threshold for a PAUSE frame helps establish the appropriate minimum amount of buffer space required to obtain lossless behavior. If insufficient buffers are reserved for PAUSE absorption, some packets may be dropped. However, if a PAUSE is injected too early, that may affect the overall utilization of the link. Link utilization is a function of the number of packets that can be on the wire at any point in time. This document does not cover link utilization considerations, since at this time the PFC implementations do not allow configurations that can influence this parameter.

Maximum Distance for a Deployed Solution

The preceding discussion outlines all the elements that need to be considered when configuring devices for lossless Ethernet and PFC. The number of variables and their interactions are too complex to allow them to be manipulated directly by end users.

The Cisco Nexus 5000 Series software currently comes preconfigured with parameters that are optimized for PFC with cable lengths up to 300 meters. However, the final determination of the maximum distance supported must take into account the devices at both ends of the wire. So if a user connects two Cisco Nexus 5000 Series Switches to each other, the 300 meters maximum distance will be granted. However, if a Cisco Nexus 5000 Series Switch is connected to a Menlo-based CNA, then you need to consider the maximum distance supported by the CNA as well, and only the minimum value should be accepted as the actual maximum distance supported by that solution. Menlo-based CNAs are preconfigured for a maximum PFC distance of 50 meters, and therefore a deployment of Cisco Nexus 5000 Series Switches connected to servers through CNAs should be built in strict observance of a maximum distance of 50 meters between any server and switch, regardless of the transceivers chosen. This distance is the maximum for one hop of the implementation and does not have direct implications for the multi-hop maximum distance between two end devices.

FCoE

FCoE is the first real application of PFC. In the context of FCoE, PFC enables a switching device to comply with the requirements of traditional Ethernet-based applications, which expect drops as a way of dealing with congestion, as well as the requirements of the Fibre Channel FC-2 layer, which assumes a lossless medium for its payloads. Both types of traffic can coexist, as long as they are classified on different CoSs, with PAUSE enabled on the FCoE CoS.

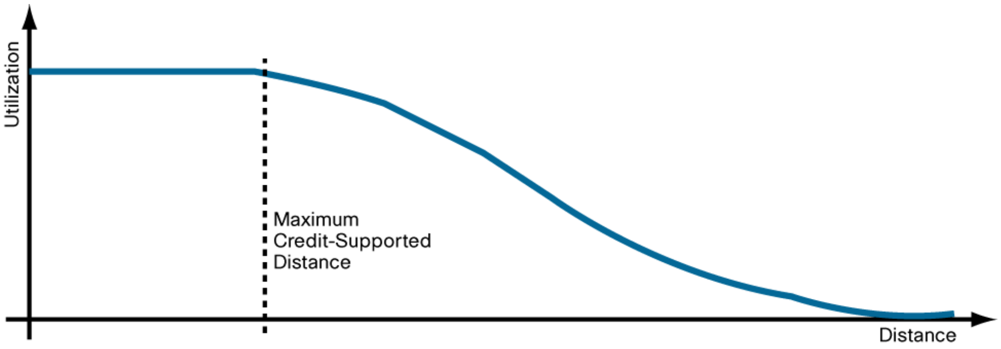
PFC and Buffer-to-Buffer Credits

How does PFC compare to Fibre Channel buffer-to-buffer credits? Both are means of achieving a lossless medium, but they address the problem in significantly different ways. Buffer-to-buffer credits use a preshared knowledge of the number of buffers available on each end, so that a sender can inject the exact number of packets that saturates the receiver buffers, and stop right then, without any explicit feedback from the receiver. The receiver needs to notify the sender when those buffers become free as the packets are drained so that both ends can keep a consistent view of the available buffers on the receiver side. These notifications are sent in the form of a receiver-ready ordered set and consume 4 bytes of bandwidth per buffer.

On the Ethernet side, ordered sets are not normally available primitives, and all processing is performed at the frame unit, a unit with a minimum size of 64 bytes. Obviously, trying to mimic Fibre Channel and consuming 16 times as much bandwidth would not be an efficient solution, and therefore Ethernet chooses a different approach. Instead of keeping the free buffer count in sync between sender and receiver, Ethernet uses a form of feedback that requires explicit communication only in response to low-buffer conditions. This solution saves bandwidth and does not require the sender and receiver to be in sync as to the count of free buffers, but its efficiency suffers as distances increase.

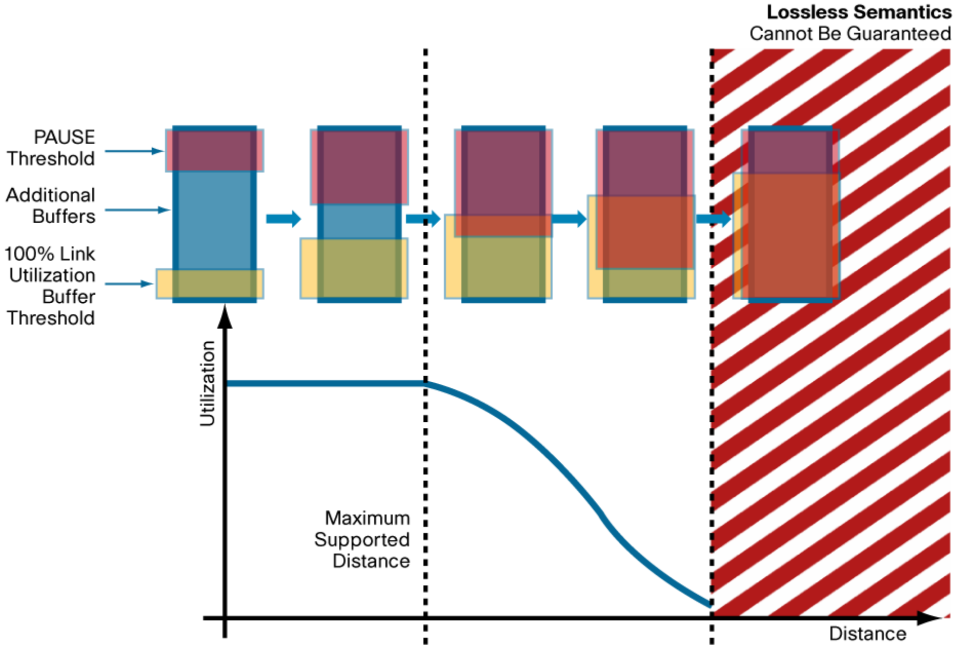
You can compare buffer-to-buffer credits to PFC from a link utilization perspective, keeping a fixed buffer configuration and extending only the distance between the two end points. Buffer-to-buffer credits are very predictable: as the distance increases, link utilization decreases asymptotically to zero, but the lossless properties of the link can be maintained (Figure 3).

Figure 3. Link Utilization with Buffer-to-Buffer Credits



PFC behaves very differently. First, you have to deal with a threshold: the further away you place the two endpoints, the larger the set of buffers you need to reserve for packet absorption after a PAUSE frame has been sent. You can reduce the threshold to maintain correctness (no drops), but this has the side effect of increasing the frequency of PAUSE generation, since a lower threshold is reached more often (Figure 4).

Figure 4. Link Utilization with PFC



More frequent PAUSE generation implies lower link utilization, similar to what happens with buffer-to-buffer credits. However, as you continue moving the two end devices further apart and continue lowering the threshold in the process, you reach a point at which the entire pool of buffers must be dedicated to absorption, and the only way to maintain correctness is to assert PAUSE at all times. Thus, at a very specific finite distance (a function of the total receiver buffers available), PAUSE becomes unable to guarantee the lossless semantics, a condition that could not occur with buffer-to-buffer credits. This limitation does not have any effect in real life scenarios in which PFC is used: trying to extend distances beyond conditions that cause very low link utilizations does not represent a meaningful use case. Besides, the problem can be resolved simply by adding more buffers to the implementation; the only intention of this section is to highlight the differences in behavior between PFC and buffer-to-buffer credits, so that users are aware of the risks in choosing to ignore clearly published distance restrictions, since these risks are different for FCoE and native Fibre Channel.

Conclusion

This document described PFC and showed how PFC is used to build a reliable Layer 2 infrastructure for the purposes of FCoE, or any application that might take advantage of this function in the future. PFC is affected by the length of the Ethernet segment between a transmitter and a receiver, and tuning the parameters properly is critical for the function to work correctly. For this reason, current implementations do not offer tuning to end users. This document also showed how the Cisco Nexus 5000 Series addresses the distance requirements and discussed how supported ranges for that platform could be extended.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Printed in USA C11-542809-01 09/15