Extreme Networks:

Congestion Management and Buffering in Data Center Networks

A SOLUTION WHITE PAPER

WHITE PAPER

Congestion Management and Buffering in Data Center Networks

TABLE OF CONTENTS

Abstract

Data center networks are undergoing a significant paradigm shift. Several technology and business trends are driving this transformation. Server and storage virtualization is changing the way applications, servers, and storage are deployed. 10G/40G Ethernet, Data Center Bridging (DCB), and Low Latency Ethernet are opening up the possibility of an all Ethernet data center that moves away from Fiber Channel and Infiniband.

Growing East-West traffic patterns within a data center is driving flatter network architectures that are optimized for fewer hops, lower latency and mesh-type connectivity. Within this changing landscape, the issue of traffic management and congestion management within the data center takes on importance as traditional approaches to congestion management such as adding arbitrarily large buffers in the network switches may potentially be detrimental to key IT requirements such as performance, latency and responsiveness of applications, as well as significant additional cost.

Buffering in Network Switches

The traditional approach to addressing congestion in networks has been to increase the packet buffers in the network switches. In traditional multi-tiered networks, typically the core network switch would have very large (or deep) buffers while the access or edge switches would have smaller (or shallow) buffers. Several reasons have led to this. An important historical reason is that in many cases, the core switch or router that was used for multi-site connectivity was the same data center core switch and/or the campus core switch. As such, the buffer sizing had to take into account worst-case long distance latencies associated with slower, long distance, Wide Area Network (WAN) links. Another consideration was that predominantly North-South traffic in client-server environments created congestion points at these core switches as relatively thin pipes exited the data center. Furthermore, the core switch would be 2-3 tiers removed from the source of the traffic (i.e. the application servers), and several hops removed from the destination (i.e. the clients) and vice versa. While transport protocols such as TCP have a built-in mechanism for congestion management, due to the number of hops between the source and destination, the time taken to relay congestion information to the TCP stack at the source (either through Active Queue Management (AQM) such as RED/ECN or through duplicate or dropped acknowledgement frames (ACKs), would be large enough that significant buffering at the congestion points in the network would be warranted.

Traditional rule of thumb has been to size buffers based on the bandwidth-delay product, i.e.:

$$B = C * RTT$$

where B = buffer size for the link, C = data rate of the link, and RTT = Average Round Trip Time.

However, research[1] suggests that while this metric may hold true in the case of a single long-lived flow, most network switches and routers typically serve a much larger number of concurrent flows. The mix of flows includes short-lived as well as long-lived flows. In such an environment the research paper shows that a link with 'n' flows requires buffer sizing no more than:

$$B = (C * RTT) / \sqrt{n}$$

The implications of this research on buffer sizing are enormous. For example, a 10Gbs link that may be a congestion point on an RTT of 250ms and 50,000 flows requires only 10Mb2 of buffering.

However, the indiscriminate addition of buffers in network switches is now unfortunately proving to be a source of congestion in networks. This phenomenon referred to as "Bufferbloat"3 has now become an area of active discussion and research. Bufferbloat can actually lead to increased congestion and TCP "goodput" (the throughput the application sees) collapse. While the reasons for this are complex, the simple way of understanding this is that TCP congestion management capabilities rely on getting quick feedback (either through duplicate ACKs, or through AQM such as RED/ECN) to adjust the congestion window at the source of the traffic and control data flow at the source. Arbitrarily large buffers leads to packets being held in the network switch's packet buffer queue for too long, causing TCP to respond far slower and perhaps too late to congestion in the network. This has the effect of opening up the congestion window, incorrectly overestimating the pipe size and leading to more traffic from the source, particularly at a time when the network is already congested. As stated in the above paper:

> *"A link that is 10 times over-buffered not only imposes 10 times the latency, but also takes 100 times as long to react to the congestion."*

Today's data center networks are characterized by low latency, high throughput, and increasing East-West traffic flows (such as VM/server to VM/server and VM/server to storage arrays). These networks have very low Round Trip Times (RTTs) due to the flatter network architectures with fewer hops between servers, and moving to high-speed 10Gbs connectivity from server to network and 40Gbs in the core of the network. In this environment, a few new considerations come into play. Notably, the need to reduce network latency between VMs/applications is becoming important.

Additionally when dealing with large, complex data sets that are constantly changing, as well as in environments such as online transaction processing (OLTP), there is greater emphasis on minimizing "stale" data being held/ or buffered in the network. Simply throwing more network buffers in these environments for the purpose of managing congestion can be detrimental for several additional reasons:

- Over-buffering in the network can lead to increased application latency and jitter, resulting in less deterministic and unpredictable application performance.
- Given the smaller RTTs in these networks (in many cases from 10s of microseconds to 100-200 microseconds), arbitrarily large buffers can actually introduce a lag between the time congestion is encountered and when the application transport protocols such as TCP detects and responds to congestion, potentially resulting in worse performance.
- Deep buffering in the network increases the amount of "stale" data being held in the network.
- Arbitrarily large buffers in network switches can significantly add to the system cost and operational complexity. This takes on greater significance as networks approach 10/40GbE speeds and port densities increase in data center networks. Building a network switch that has large off-chip packet memory buffers adds significant cost to the switch due to the complexity of memory controllers required to address both very high bandwidth and very high density.

At the same time, however, adequate buffering in the network is critical to address situations such as microbursts and incast type scenarios. Incast refers to a many-to-one traffic pattern commonly found in scale-out distributed storage, Hadoop, and big data application networks. In an Incast scenario, a parent node typically places a barrier synchronized request for data to a cluster of nodes. As a result, the cluster of nodes simultaneously reponds to this request, resulting in a micro-burst of traffic from many machines to the parent node. If any of these responses are lost, the parent node is forced to wait for TCP to timeout at the sender nodes before the re-transmission happens. Additionally, the TCP windows back off, leading to degraded throughput. See Figure 1 below. Incast is also commonly referred to as "TCP Incast" since many of these applications use TCP, leading to application throughput collapse.

While the subject of addressing TCP Incast throughput collapse is an active research subject, it is clear that the network switches need to have adequate burst handling collapse is an active research subject, it is clear that the capability to absorb traffic bursts such as from Incast-type scenarios, and to address periods of temporary congestion in the network. Network traffic is inherently bursty and providing the right buffering in the network can help smooth this out and improve application performance. Care must be taken, however, to not overprovision the buffer pool as this can lead to lethargic application performance and can cause transport protocols such as TCP to respond poorly to congestion in the network.

Congestion Management in Data Center Networks

While the subjects of active congestion management, burst handling capability, TCP Incast and bufferbloat are all complex issues, broadly two key aspects need to be considered:

- Ensuring that the network switches have good burst absorption capabilities, while at the same time they do not add excessive latency, jitter, and unresponsiveness to the application.
- Ensuring that congestion in the network is signaled back to the source quickly so that the source can take remedial measures quickly so as not to continually congest the network.

These are explored below:

BURST ABSORPTION AND BUFFERING IN NETWORK SWITCHES

When considering buffering in network switches, there are two aspects to consider. The first, buffer allocation strategy, and the second, optimal buffer size.

BUFFER ALLOCATION STRATEGY

A buffer allocation strategy that allows buffers to be utilized as a central buffer pool from which congested ports can draw on during periods of congestion provides a far more effective form of packet buffer memory than one that is allocated statically across all ports.

An important aspect of having a shared buffer pool is the ability to provide per-queue dynamic adaptive thresholds. What this means is that even with the shared buffer pool, the queue drop thresholds for each queue are not statically assigned. Rather, the queue drop thresholds adapt dynamically to the transient congestion conditions. This ensures that queues or ports that experience congestion can draw on the buffer pool more extensively than they otherwise could with a static queue drop threshold if more buffers are available in the system. At the same time, however, if multiple queues are experiencing congestion, or in other words, if the buffer pool is running low, then the queue drop thresholds get dynamically readjusted to limit how much an already congested queue can draw on the buffer pool so that other queues do not get starved and get their fair share as well. This dynamic threshold on a per-queue basis is a key advancement in buffering technology and one which enables not just very good burst absorption capabilities but also fairness among queues[3]. The dynamic threshold provides a single parameter that can be tuned to adjust the buffer allocation strategy across all queues and congestion conditions. Simulations4 have shown that for a switch with 64 10GbE ports, using a uniform random distribution across all ports, a burst size of 32KB, packet size of 1500 bytes, a loading factor of 80 percent on all ports, and a target frame loss rate of 0.1 percent (to guarantee high TCP goodput), a static per-port packet buffer allocation scheme would require 26.5 MB of packet memory to achieve the target frame loss rate. By contrast, if the buffer pool allocation is dynamic with adaptive thresholding, (see section on Smart Buffers below), the same packet loss rate restrictions can be achieved with a packet buffer of 5.52MB. In other words, a switch using per-port static packet memory allocation could require up to five times as much packet memory as a switch using dynamic shared packet buffer pool with adaptive thresholds, to achieve equivalent burst absorption and frame loss performance. See Figure 2 below.

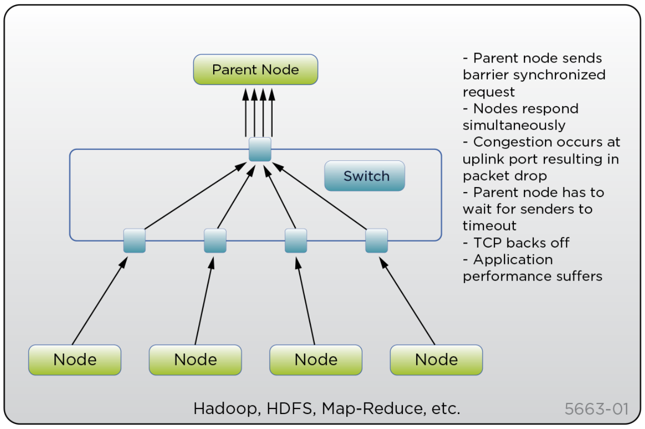

Parent Node
Switch
Node
Node
Node
Node
- Parent node sends barrier synchronized request
- Nodes respond simultaneously
- Congestion occurs at uplink port resulting in packet drop
- Parent node has to wait for senders to timeout
- TCP backs off
- Application performance suffers
Hadoop, HDFS, Map-Reduce, etc.
5663-01

Figure 1: TCP Incast Scenario

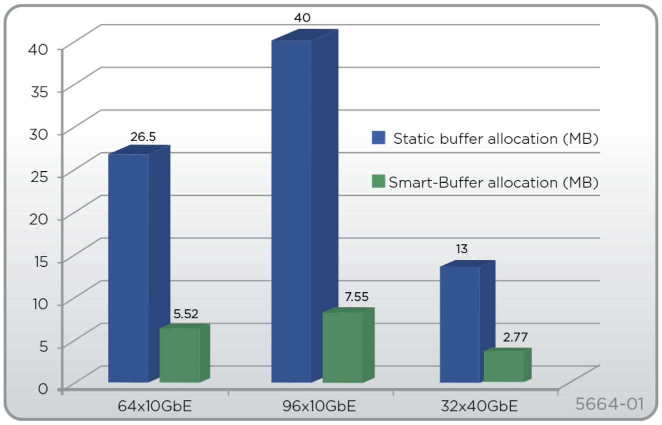


Figure 2: Smart Buffer Delivers up to 5 Times Better Packet Buffer Utilization

Another benefit of using adaptive dynamic buffering technology is in the use of multicast traffic. Typically, multicast packets coming in on one port need to get replicated across multiple egress ports. This generally has two implications. The first is that it significantly increases buffering requirements due to the need to hold multiple copies of the packet. The second is that it can potentially lead to larger variances in the time when the copies get transmitted out. However, by using smart buffering technology, the packet need not consume multiple copies in the packet memory. Rather, a single copy of the packet is maintained in the packet buffer with multicast packet modifications directly carried out in the egress pipeline. This significantly reduces the packet buffer size, and just as importantly, the replication latency is very consistent across multiple egress ports. This latter issue is particularly important in financial exchange points and trading floors where data, transactions and feeds need to be distributed to customers with minimal variance wherever multicast is used in order to ensure neutrality. Simulation results in Figure 3 below show a single multicast stream at 100% line rate sent in on port 1, being replicated across five ports on the same switch silicon. All ports in the simulation are 40GbE ports using Quad Small Form-factor Pluggable (QSFP). As can be seen, at any given packet size the latency variance across egress ports 2, 3, 4, 5, 6 is almost negligible. All latency numbers below are in microseconds.

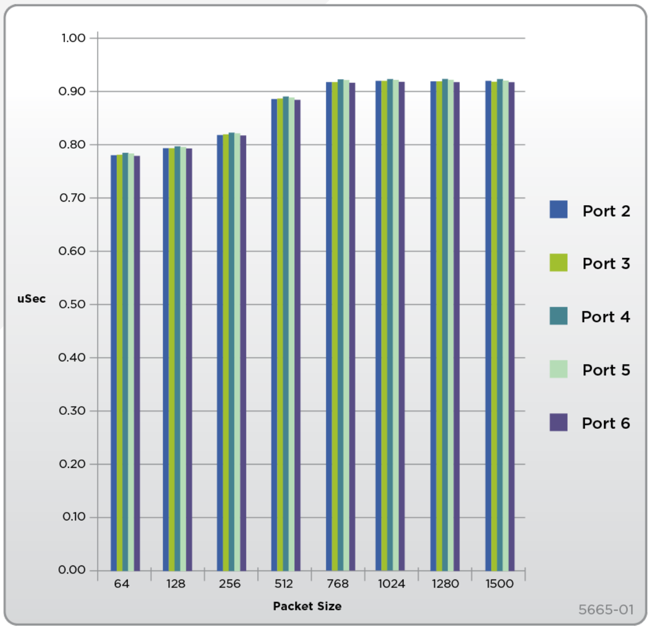


Figure 3: Multicast Latency Variance across Ports using Smart Buffering

Data center products from Extreme Networks® such as the Summit® X670 and the BlackDiamond® X8 utilize dynamic adaptive smart buffering technology which can provide very efficient buffer utilization, great burst absorption capability, ultra-low latency, and very predictable multicast packet replication times across ports. ExtremeXOS®, which powers all of Extreme Networks switching products, provides users the ability to fine-tune the burst absorption capability by a simple adjustment of the discard thresholds as a percentage of the total buffer pool. These values can be customized on a per-port basis, so for example on an uplink port where you could expect congestion, the adaptive discard thresholds could be set higher. In addition, ExtremeXOS also provides per-port and per-queue congestion statistics to provide users insight into the network bottlenecks.

BUFFER SIZING

The subject of buffer sizing is a complex one, governed by several different factors such as TCP window size, including receive and congestion windows, network RTT, cut-through versus store and forward switching, traffic patterns within the network, etc. In all cases, however, sufficient buffering needs to be provided in the network switch to absorb short bursts and address temporary congestion. However, the buffering needs to be sized correctly so as to not introduce too much delay or too much lag for transport protocols to adapt to more sustained network congestion conditions. For example, in many of today's data center top of rack (TOR) or access switches, it is common to have 48 ports for server connectivity with the rest of the ports on the switch for uplinks. In one possible worst-case scenario, it is possible that all 48 servers simultaneously burst traffic towards one single uplink port. If the burst is a temporary short burst, then the network switch needs to have adequate buffering to absorb such a burst. If the burst is not a short burst but a more sustained traffic pattern, then at some point the network will become congested and the applications will need to throttle back. This will happen when the congestion feedback reaches the source in a timely manner and is the subject of the Congestion Management section below.

Taking the case of where the bursts are short-lived microbursts, if for example the burst size were 64Kbytes (65536 bytes) from each server (which is the default initial TCP receive window that is communicated to the sender in many TCP implementations), the switch would need to have a buffer capacity of 48ports* 64KBytes=3.072MBytes to be able to absorb that burst fully, assuming the outgoing link is congested and the entire burst from every source would need to be buffered.

The above example is illustrative of the buffer sizing needed in addressing micro-bursts from several sources simultaneously. While several other factors can come into play here such as TCP window scaling, the size of the initial congestion window, randomization of RTOs, etc., the example above provides a good insight into the packet buffer sizing requirements for addressing microbursts. In reality, packets are stored in silicon using fixed-sized buffer pools, the size of which varies from silicon to silicon. So, for example, if the internal buffer pool is divided into 256 byte buffers, and the 64KB bursts from each source are using 1500 byte packets, then each packet would consume 6 internal buffers, leading to 36 bytes wasted per packet: 256bytes*6 = 1536 bytes. A 64KB burst would lead to 43 1500 byte packets, with a last packet being 1036 bytes. This last packet would result in an additional 12 bytes wasted in terms of internal 256 byte buffers. In effect, a 64KB burst of 1500 byte packets would result in an additional 36*43+12=1560 bytes. With 48 ports all bursting together, that's an additional 48*1560bytes = 74880bytes. Therefore, the total packet buffer size required would be 3.072MB+74880B=3.145MB.

While the above calculation is an example of buffer sizing required to absorb a temporary burst, several other factors come into play. Simulations (see Figure 2 above) have shown that for a switch with 64 10GbE ports, using a uniform random distribution across all ports, a burst size of 32KB, packet size of 1500 bytes, a loading factor of 80 percent on all ports and a target frame loss rate of 0.1 percent (to guarantee high TCP goodput), a packet buffer of 5.52 MB was shown to be effective in achieving the target frame loss rate2. The results are based on adaptive dynamic sharing of buffers (i.e. not a static per-port allocation).

Extreme Networks Summit X670 TOR switch provides 9MB of smart packet buffer across 48, 10GbE and 4, 40GbE ports. Similarly, the Extreme Networks Black Diamond X8 utilizes shared smart buffering technology both on its I/O modules as well as on its fabric modules or example, the 24-port 40G module uses four packet processing silicon chips, each of which provides 9MB (72Mb) of smart packet buffer. A small number of buffers are reserved on a per-port/per-queue basis, which is configurable. The rest of the buffers are available as a shared pool that can be allocated dynamically across ports/queues. As described earlier, adaptive discard thresholds can be configured to limit the maximum number of buffers that can be consumed by a single queue. The threshold adjusts itself dynamically to congestion conditions to maximize the burst absorption capability while still providing fairness. As shown in the simulation results, the 9MB (72Mb) of shared dynamic smart buffers provides excellent burst absorption capabilities, but are still not bloated to a point where they serve as a detriment in signaling congestion feedback to the end points.

CONGESTION MANAGEMENT

Right sizing buffers in the network along with the right buffer management algorithm ensures that the network can absorb temporary bursts of traffic. Equally important is that sustained congestion in the network is signaled back to the source quickly for two reasons:

- The first is that the source can adapt its transmission rate to match the network conditions. For example, TCP has sophisticated congestion management capabilities which rely on congestion feedback from the network either implicitly (duplicate ACKs or timeout) or explicitly (through AQM such as RED-ECN).
- The second is that the larger buffers at the end station can be utilized for buffering the source's traffic rather than in the network so that the traffic management and buffering is at the granularity of the source's flow rather than at an aggregate level within the network.

Today's data center networks are designed with fewer hops, which can provide faster feedback to the source. Congestion in the network can be signaled back to the source at Layer 2 or Layer 3/4. Layer 2 signaling takes advantage of the new Data Center Bridging (DCB)-based, Priority-based Flow Control (PFC) capability. Layer 3/4-based signaling takes advantage of transport protocols such TCP's built-in congestion management capability.

LAYER 2-BASED CONGESTION MANAGEMENT: DCB-PFC CAPABILITY

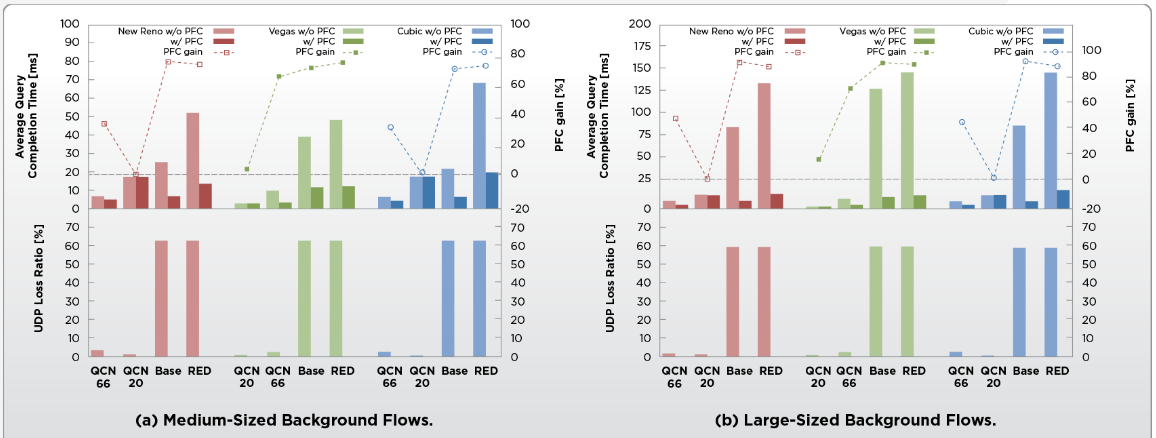
One of the mechanisms to "push" congestion information back towards the source is to leverage PFC. With PFC, the network is effectively partitioned into virtual lanes, each lane representing a traffic class. For example, storage traffic such as iSCSI/NFS/FCoE may be assigned to its own traffic class. Streaming video may be assigned to another traffic class. When congestion for any traffic class is encountered in a network switch, the network switch generates pause frames specifically for that traffic class. The previous hop network switch that receives the pause frame will pause transmission of traffic for that traffic class on that network port by utilizing its own built-in buffers.

When its buffer pool starts filling up, the network switch generates another pause frame for that traffic class, which in turn causes the previous hop to pause traffic for that traffic class. In this manner, PFC uses back pressure to quickly propagate congestion information back towards the source, where ultimately it pauses the source from sending traffic for that traffic class. Note that traffic belonging to other traffic classes continue to flow since the pause frame is specific to a traffic class. When deployed close to the traffic sources that can contribute to network congestion, PFC provides a quick and effective mechanism to adapt traffic to network congestion conditions.

A recent study5 tested PFC (along with other technologies) in TCP-based networks. The study tested several different configurations and benchmarks using various workloads as well as various TCP implementations (New Reno, Vegas, Cubic). The results of the study state:

> *"… PFC has consistently improved the performance across all the tested configurations and benchmarks. The commercial workload completion time improves by 27% on average, and up to 91%. Scientific workloads show higher gains by enabling PFC: 45% on average, and up to 92%."*

As an example, one of the tests carried out in the study was with a commercial workload utilizing TCP, with UDP background traffic. The graphs below in Figure 4 reproduced from the above study indicate the benefit of using PFC in such an environment across all three TCP implementations. The upper graphs indicate average query completion times (smaller is better) with and without PFC for each of the three different TCP implementations.



Commercial Workload with UDP Background Traffic. The upper part of the graphs shows the average query completion time, while the lower part shows the loss ratios of the background UDP flows. Bars are grouped in three categories based on the TCP version. Within a category, bars are sorted increasing with average query completion time without PFC.

5666-01

Figure 4: Benefit of Using PFC with Different Workloads

(Reproduced from the paper "Short and Fat: TCP Performance in CEE Datacenter Networks" by Daniel Crisan et al.)

Independent testing6 of various different switch vendors' switches under congestion conditions showed that using a backpressure-based mechanism can be very effective in addressing congestion. As can be seen in the table reproduced below (Figure 5), the BlackDiamond X8 was the top performer in the congestion test category both for Layer 2 traffic and Layer 3 traffic.

Care must be taken, however, when using PFC in a multi-tiered network where congestion points are several hops away from the source. This is because using a pause-based scheme such as PFC (even where the pause is at the granularity of the traffic class) may result in congestion spreading across many hops. This may not be a problem in most new data center networks where VMs/servers are one or two hops from other VMs/servers. In other words, flatter collapsed network architectures are well suited for PFC. By using right-sized buffers and not over-buffering the network, the network becomes responsive to sustained congestion and can backpressure the source in a timely manner. At the same time, as mentioned earlier, by providing a dynamic shared buffer pool with adaptive thresholding, the network switches provide great burst absorption capabilities to deal with temporary bursts or temporary congestion.

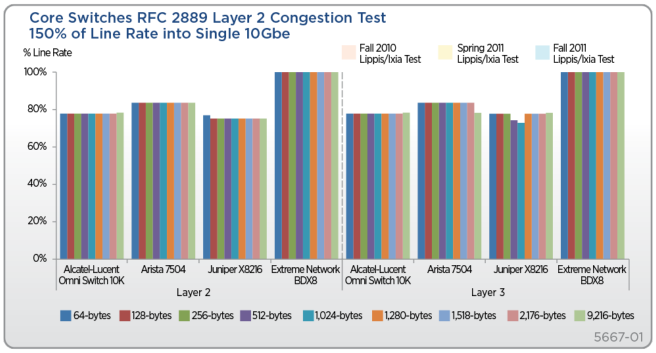


Figure 5: Core Switches RFC 2889 Layer 2 Congestion Test

PFC and DCB in general are relatively newer technologies. However, most new 10GbE Converged Network Adapters (CNAs) and NICs, and most new 10/40G Ethernet switches support DCB today. Both the Summit X670 and the BlackDiamond X8 series switching products support PFC (along with other DCB capabilities such as Enhanced Transmission Selection (ETS) and Data Center Bridging Exchange (DCBX).

LAYER 3/4-BASED CONGESTION MANAGEMENT

Layer 3/4-based congestion management utilizes congestion management capabilities built into transport protocols such as TCP to provide congestion management at the granularity of the individual flow. TCP detects congestion either implicitly through duplicate ACKs or Retransmission Timeouts (RTO), or explicitly through mechanisms such as ECN (Explicit Congestion Notification).

The area of TCP congestion management in data center networks is an active area of research. TCP's default minimum RTO in many TCP implementations is set at 200ms. This is an artifact of the fact that this number was optimized for WAN delays. However, many data centers have RTTs in the 10s or 100s of microseconds range. There is active research ongoing that suggests the use of High Resolution Timers in the range of several 10s or 100s of microseconds for TCP for the purpose of estimating RTT as well as for setting the minimum RTO. The question of network buffer sizing for data center networks with 10s or 100s of microsecond RTT is a complex one as well. On the one hand, with newer research7 suggesting that use of microsecond timers can help improve performance in data center networks, sizing network buffers that do not artificially skew TCP's measurement of RTT is important. For example, with 100us RTT and 10Gbs Ethernet links, a network buffer allocation based on the traditional BDP (Bandwidth Delay Product) i.e.

B = C * RTT (see section 2 above) results in about 1Mb buffer (10Gbs * 100us). A cluster of 40 servers all bursting a maximum of the BDP worth of data at 10Gbs speed simultaneously would result in a burst of 40Mb or 5MB, which would suggest that having adequate network switch buffers to absorb such a burst should be sufficient. The number is further reduced significantly if the modified formula of B = (C * RTT)/√n (see the Buffering in Network Switches section above).

In addition to RTO or duplicate ACK-based mechanisms, other approaches that explictly signal congestion such as ECN and propagate congestion information quickly to the source are also possible and being actively pursued. These build upon Active Queue Management (AQM) in conjunction with transport protocol congestion management capability. For example, utilizing RED/WRED AQM, congestion can be indicated by marking the CE (Congestion Experienced) code point in the IP header. A TCP receiver can then use this to signal congestion back to the sender by use of the ECE (ECN-Echo) flag in the TCP header. Use of ECN in the past has been very limited due to buggy implementations. However, there is renewed interest in utilizing ECN capability in Data Center Networks. For example, Data Center TCP (DCTCP)8 is one approach that holds a lot of promise. DCTCP utilizes the ability of newer network switches to indicate congestion by marking the CE (Congestion Experienced) code point in the IP header of packets. This is then used in conjunction with TCP at the receiver to accurately convey congestion information back to the sender. Based on this, the sender adapts its window to match the congestion conditions. DCTCP has been shown to significantly benefit performance in data center networks across a variety of workloads and traffic loading conditions. Utilizing right-sized shared buffers (as opposed to arbitrarily large buffers) can actually benefit DCTCP as it allows timely feedback of congestion information back to the sender. DCTCP is relatively new and implementations are just beginning to come on the market. For example, Windows Server 2012 will have support for DCTCP. Additionally, DCTCP is also available as a patch to Linux 2.6.38.3.

Extreme Networks data center products support AQM such as RED/WRED. Additionally, the products are capable of supporting marking for ECN, which can be enabled in the future through a software upgrade.

Smart Buffering Technology

As indicated earlier in the document, data center products from Extreme Networks utilize smart buffering technology (from Broadcom). Smart buffering technology utilizes on-chip packet buffer memory that minimizes latency by avoiding packet reads and writes to and from external memory. On-chip packet buffers are dynamically shared across all ports/queues and can provide excellent burst absorption and optimal buffer utilization. A key aspect of the smart buffering technology is that queue discard thresholds are sized dynamically based on congestion conditions, or in other words, the remaining unused buffer pool. By dynamically adapting to congestion conditions, longer, more congested queues are prevented from taking undue buffers away from shorter or less congested queues. However, during periods of relatively low congestion, a queue can take up far more buffers in order to absorb a temporary burst of traffic. This dynamic adaptation of queue discard thresholds is referred to as Fair Adaptive Dynamic Threshold (FADT). Research9 has shown that FADT provides fair access to buffer resources, and as a result provides very good port throughput isolation. Furthermore, since FADT adapts to congestion dynamically, no parameter tuning s needed based on differing congestion situations. Smart buffering technology based on FADT can provide the following benefits:

- **Excellent Burst Absorption** – This ensures that during transient congestion the buffer management algorithm can provide enough resources to enqueue packets leading to fewer frame drops.
- **Fair Shared Buffer Pool Access** – This ensures that during times of congestion, uncongested ports do not get starved out of access to the shared buffer pool.
- **Port Throughput Isolation** – This ensures that during times of congestion the buffer management policy does not unfairly throttle link utilization on uncongested ports. Traffic Independent Performance – Since congestion conditions vary dynamically in time and with different workloads, it is important to ensure that the selection of parameters for a buffer management policy should not depend heavily on the nature of traffic. In other words, the buffer management should provide optimal performance for a wide range of traffic scenarios with minimal parameter tuning.

Additionally, several improvements in multicast handling in conjunction with smart buffering technology provide significant benefits. Multicast packets go through the MMU (Memory Management Unit) as a single packet. In other words, there is a single copy of a packet in the packet memory no matter how many replications are needed. The MMU replicates the packet out of the single copy to each of the egress ports and possibly multiple times on each egress port. If any packet modifications are required, the egress pipeline performs packet modifications on the fly. There is no additional packet buffer required for packet modifications. This technology provides some key benefits:

- Packet buffer sizing can be kept to a minimum with no performance impact in the presence of multi-destination traffic. In other words, mixing unicast and multicast traffic does not lead to significant incremental buffer consumption.
- Since all packet modifications are done in the egress pipeline, multicast latency is very low.
- Additional latency variance across egress ports is also kept to a minimum.

Practical Application Examples of Smart Buffering Technology

In general, smart buffering technology provides a solid foundation for any Ethernet packet switch. By minimizing buffer requirements for both unicast and multicast traffic as described above, costs can be kept low and the switch can still provide excellent burst absorption capabilities to address temporary periods of congestion. Equally important, as port density and port speeds increase, the cost and complexity associatedwith large off-chip buffers can be eliminated.

High Performance Computing (HPC) applications typically require a fast, low-latency, low-jitter fabric for interconnection of compute clusters. In these applications, simply providing a high capacity interconnect fabric is not sufficient. In other words, high bandwidth does not equal low latency. Several factors contribute to latency. Store and forward modes of packet handling within the switch add latency. Storing and reading packets in and out of external off-chip packet memory adds latency. Large buffers add to jitter and latency variance. And Ethernet's best-effort nature contributes to packet loss and unpredictable performance. In these HPC applications, a switch that supports cut-through forwarding and on-chip smartsized buffers provides the right approach to addressing the low latency requirement. FADT capability ensures that in big data applications employing map-reduce or disk striping-type functionality, performance is not compromised. Having on-chip smart buffers also allows for very high density wire speed (at 10/40/100Gbs) switches, minimizing both the cost of the individual switches as well as the number of switches required for the interconnect. Finally, the use of smart buffering in conjunction with DCB and in particular PFC and ETS can provide a more predictable and more reliable Ethernetbased transport for such applications.

Similarly, financial trading network deployments have their own set of characteristics. One attribute of financial trading exchanges is their order processing latency. In these networks, switching latencies at each hop are in the very low single-digit microsecond or hundreds of nanosecond range. Another key requirement is very low latency variance. In other words, consistency in the latency of order processing is a key requirement. Additionally, these networks tend to be heavy users of multicast traffic. Multicast is used for a variety of reasons, including distributing financial ticker information to clients. One challenge when using multicast for distributing trading information to clients is in ensuring that all clients get access to the information at the same time. In other words, neutrality is a big issuein financial trading floors. In all these scenarios, smart buffering technology plays a big role. As mentionedabove, cut-through forwarding and right-sized on-chip buffering can reduce the latency and latency variance. Additionally, as described above, the use of a singlecopy multicast can significantly reduce the buffer requirements in the presence of multi-destination traffic. And finally, replicating the packet directly on the egress of each port produces very little latency variance across ports, which addresses the neutrality considerations for this market.

While smart buffering technology with FADT has broad applicability, the above examples are representative of how smart buffering technology addresses some of the most stringent requirements in very specific markets.

Summary

The subject of network switch buffering is a complex one with a lot of research ongoing on this subject. Traditional approaches of over-buffering in network switches may actually be a cause of performance degradation, where performance could be defined as latency or application-level throughput ("goodput") along with adding significant cost to the network infrastructure. A combination of dynamically allocated right-sized shared smart buffers, along with active congestion management capabilities can provide both good burst absorption capabilities to address temporary congestion, along with responsive feedback systems for timely end-to-end congestion control.

- As an example, Extreme Networks data center switch products (such as the Summit X670 and BlackDiamond X8) support on-chip dynamic shared buffer allocation schemes along with DCB and AQM. This can enable:
- Significantly lower buffer sizing requirements compared to static per-port schemes
- Excellent burst absorption capability
- Very low latency in both cut-through and storeand-forward modes
- Very low multicast replication variance across ports
- Very efficient buffer utilization for multicast traffic with single copy replication
- Improved application performance and better network congestion management in conjunction with PFC

Bibliography

1. Sizing Router Buffers. G. Appenzeller, I. Keslassy and N. McKeown. 2004, Proceedings of the SIGCOMM.

2. Bufferbloat: Dark Buffers in the Internet. J. Gettys, K. Nichols. 2012, Communications of the ACM, Vol 55 No. 1, pp. 57-65.

3. Dynamic queue length thresholds for shared-memory packet switches. A. K. Choudhury, E.L. Hahne. s.l. : ACM Transactions on Networking, 6, 1998.

4. Das, Sujal and Sankar, Rochan. Broadcom Smart-Buffer Technology in Data Center Switches for Cost Effective Performance Scaling of Cloud Applications. [Online] 2012. http://www.broadcom.com/collateral/etp/SBT-ETP100.pdf.

5. Short and Fat: TCP Performance in CEE Datacenter Networks. Daniel Crisan, Andreea Simona Anghel, Robert Birke, Cyriel Minkenberg and Mitch Gusat. s.l. : Hot Interconnects, 2011.

6. Lippis, Nick. Lippis Report: Open Industry Network Performance and Power Test for Cloud Networks - Evaluating 10/40GbE switches Fall 2011 Edition. 2011.

7. Safe and Effective Fine-grained TCP Retransmissions for Datacenter Comm. Vijay Vasudevan, Amar Phanishayee, Hiral Shah, Elie Krevat, David G. Andersen, Gregory R. Ganger, Garth A. Gibson, Brian Mueller. s.l. : SIGCOMM, 2009.

8. Data Center TCP (DCTCP). Mohammad Alizadeh, Albert Greenberg, David Maltz, Jitu Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta , Murari Sridha. s.l. : SIGCOMM, 2010.

Extreme networks

http://www.ExtremeNetworks.com/contact / Phone +1-408-579-2800