The Role of Buffer Management in Controlling the Effects of Congestion over 10 GbE Links

Abstract

This white paper describes the rationale for providing switch/routers with port buffers that are scaled to the expected delay-bandwidth product (DBP). DBP-sized buffers, together with RED congestion avoidance, can be used to optimize the utilization of congested links by TCP applications.

While large buffers primarily benefit long-term TCP application flows, QoS functionality can be leveraged to allocate appropriate levels of buffering and bandwidth to different classes of traffic and to protect delay-sensitive applications from excessive queuing delays associated with large buffers.

Next the document focuses on the general requirements for optimizing buffer management for 10 GbE matrix switches. This discussion is illustrated by a description of how the architecture and design of the Force10 E-Series of switch/router leverage the combination of line-rate packet processing and QoS features to optimize buffer management for 10 Gigabit Ethernet (GbE) and eventually the next generation of Ethernet speeds.

One premise holds true. There is no such thing as a network whose end-to-end paths are guaranteed to be congestion free.

In the Internet, virtually all traffic from ISP subscribers aggregates at the uplinks to the core. Therefore, as long as the utilization of the uplink is low and the utilization of the core links is low, congestion is infrequent enough to allow delay-sensitive Internet telephony applications based on SIP and Skype to work quite well, even across parts of the Internet where no QoS is configured. Problems that do arise for these applications are primarily due to congestion at aggregation points in the core of the network. Congestion can still occur in spite of the fact that the average utilization of core links is generally in the range of 10-30%.

Congestion is considerably more prevalent in the enterprise network because:

- access bandwidths are higher (especially with the trend toward Gigabit Ethernet to the desktop and server connections beginning to move to 10 GbE)
- traffic patterns are less predictable
- uplinks (as well as access links) are frequently intentionally oversubscribed due to economic necessity

Over time, it is expected that congestion will become more prevalent in the Internet. This will occur as high bandwidth applications (including peer-to-peer applications) become more popular, and the average access bandwidth continues to increase rapidly. As these changes take place, it is likely that congestion will start to more severely affect packet loss-sensitive applications and delay-sensitive applications such as VoIP.

The first line of defense against congestion is to have adequate buffering available. Adequate buffering is necessary to minimize packet loss and to maximize the utilization of the end-to-end network. The prevalent delay-bandwidth product rule of thumb is that a router port should have a buffer capacity B equal to the average Round Trip Time (RTT) of the TCP sessions flowing through the link times the link bandwidth (BWlink in bps).

\[ B = RTT \times BW_{link} \]

Since the propagation delays for transcontinental links and transoceanic links are large, routers and switches with high bandwidth WAN interfaces need large buffers. For example, according to the delay-bandwidth rule of thumb, a switch/router with 10 GbE WAN interface linking two research campuses 10,000 miles apart would require approximately \( 200 \text{ ms} \times 10 \text{ Gbps} = 250 \text{ MB} \) of buffering available to that interface.

In the LAN, the bandwidth delay product is obviously much lower due to low RTT. Therefore, buffer sizes can accordingly be reduced to a few ms of link capacity. However, switch/routers with fixed buffer sizes scaled to the campus LAN can take only limited advantage of high capacity MAN/WAN links, such as GbE and 10 GbE over dark fiber or SONET.
Right-Sizing Switch/Router Buffers

The delay-bandwidth product rule of thumb is based on the desire to keep a congested link on the end-to-end path as busy as possible in order to maximize the throughput of the network for long-term application flows based on TCP, such as large file transfers with FTP. TCP's congestion control algorithm is designed to continually probe the network to find the maximum possible transfer rate. This is done by deliberately attempting to fill the buffer of any congested port in order to ensure that the link is fully utilized. Buffer overflow causes packet loss that throttles back the sender's forwarding rate.

The typical behavior of a single long-term TCP session with a bottlenecked link having a buffer sized at \( B = \text{RTT} \times \text{BW}_{\text{link}} \) is shown in Figure 1. At the start of the session, the sender's congestion window size is increased exponentially (slow start) until the buffer fills and multiple packets are dropped. The lost packets provide the feedback that causes the sender to cut the previous window size to zero and recommence slow start until the congestion window is half the previous maximum and then increment the congestion window size linearly (the congestion avoidance phase) until the buffer fills again and a single packet is dropped. At this point, the sender reduces its window size by one half and reenters the congestion avoidance phase, resulting in the sawtooth pattern being repeated throughout the session. With \( B = \text{RTT} \times \text{BW}_{\text{link}} \), the bottleneck link buffer will fill and then be completely drained just at the time that the sender's packets begin arriving again, maximizing the utilization of the congested link.

If the congested link is under-buffered, the congestion window follows the same sawtooth pattern, but when the window size is halved and the sender pauses, waiting for ACKs, the buffer goes empty and the bottleneck link goes idle. If the congested link is over-buffered, the link remains fully utilized. However, when the congestion window is halved, the buffer does not completely empty, increasing the average queuing delay.

Thus, the TCP end-to-end congestion control mechanism tends to keep the network load oscillating with the size of the load oscillations determined by router buffer size. Since the loss, delay, and throughput characteristics of the network are primarily the result of these load oscillations, buffer size is a critical aspect of network design.

![Figure 1. TCP congestion window size and queue depth over time](image)

In general, increasing the buffer size reduces packet loss and increases link utilization but results in longer queues at the bottleneck link and higher end-to-end delays.

The reasoning behind the delay-bandwidth rule of thumb is based on the notion that a single TCP session might be capable of overloading a link in the core of the network. However, the same buffer sizing rule of thumb and sawtooth load behavior of a single TCP session continues to hold for a number of TCP sessions (up to 500 or more) sharing a congested link. This is true because the sessions tend to synchronize due to buffer overflow feedback being delivered to all the TCP senders at approximately the same time.

Maximizing throughput of individual TCP sessions over high bandwidth delay paths may also require experimentation with different TCP implementations, including those supported in Ethernet adapters with hardware-supported TCP Offload Engines (TOE), and careful tuning of TCP parameters. In particular, the usual recommendation is to set the host's TCP receive window (rwin) to the RTT times the bandwidth of the slowest link on the end-to-end path between sender and receiver. Therefore the host buffer sizes may need to be the same size as that of the intervening switch/router ports.

In most networks, the congested link will be carrying a combination of different types of flows, including short-term TCP flows and UDP flows in addition to long-term TCP flows (which currently tend to dominate bandwidth consumption). While large buffers do not adversely affect short-term TCP flows, the implicit queuing delay can have a negative effect on delay-sensitive UDP applications. Where delay-sensitive applications are a major consideration, QoS functionality can be leveraged to minimize queuing delay for applications such as VoIP, while still providing the large buffer space needed for optimized bulk TCP data transfers.
Congestion Avoidance with FIFO Queues

While buffer overflow provides the feedback mechanism that throttles back offered load, it would be preferable to have a less drastic means of providing the needed feedback to the sending host. This is because buffer overflow results in "tail drop" of in-transit packets, which disrupts UDP sessions and can simultaneously drive numerous TCP sessions all the way back to slowstart – a condition called global synchronization. Reducing the number of occurrences of tail drop is the rationale behind congestion avoidance mechanisms.

With no Quality of Service (QoS) features enabled, packets are buffered in a single FIFO egress queue and serviced in the order in which they are received. In the event of congestion, all subsequent packets are dropped (tail drop) after the buffer overflows.

Random Early Detection (RED) is a congestion avoidance scheme that tries to prevent the oscillations between link over-utilization and under-utilization that occur during global synchronization caused by tail drop. RED tries to control the queue depth by randomly dropping packets as buffers fill beyond a configured threshold level. The comparatively small number of affected TCP hosts will accordingly decrease their transmission rate. If the buffer continues to fill, a higher and higher percentage of packets is randomly dropped. Ideally, RED greatly reduces the number of buffer overflows, distributes packet loss over longer time intervals, and minimizes queue depth while still absorbing traffic spikes.

As shown in Figure 2, RED begins dropping packets when the average queue depth reaches a minimum threshold. Beyond this threshold the probability of a packet being randomly dropped increases linearly up to the maximum threshold, after which all packets are dropped, as in tail drop. The packet drop probabilities are calculated based on the minimum threshold, maximum threshold, and mark probability denominator. With a mark probability denominator (m), the fraction of packets dropped is 1/m when the average queue depth is at the maximum threshold. The router automatically determines the average queue size based on the previous average and the current size of the queue. The formula used is:

\[
\text{New average} = \left[\frac{\text{old average}}{1 - 2^{-n}}\right] + \left(\frac{\text{current queue size}}{2^n}\right)
\]

where \( n \) is a user-configurable exponential weighting factor.

Other versions of RED modify the drop profile in order to approach the tail drop point more gracefully. For example, with the RED implementation in Force10’s switch/router FTOS operating system, the drop profile (shown in Figure 3) is implemented through a piecewise linear algorithm. Force10’s algorithm approximates ideal quadratic behavior and eliminates the abrupt step function at the maximum threshold.

Congestion Avoidance with Multiple QoS Queues

When large buffers configured as FIFO queues begin to fill, the result is large queue depths and the long queuing delays at the point of congestion. For example, if a 125 MB buffer on a 10 GbE WAN port fills to half capacity, that would introduce a 50 ms queuing delay for any new incoming packet. Therefore, switches with large buffers need to be able to protect any critical delay-sensitive applications (or critical loss-sensitive applications) by employing QoS-based intelligent buffering in conjunction with congestion avoidance.
When QoS is enabled on the switch, the port buffers are divided into multiple individual queues. Traffic sensitive to delay and delay variance, such as VoIP packets, can be placed in a queue that receives strict priority over the remaining queues. The other traffic can be assigned to one of a group of queues that are managed by a scheduling mechanism, such as Weighted Round Robin (WRR) or Weighted Fair Queuing (WFQ). Each WRR or WFQ queue can be configured with one or more drop thresholds. The combination of multiple queues within a buffer, and the drop thresholds associated with each queue, allow the switch to make intelligent decisions when congestion occurs.

When RED is used in conjunction with multiple traffic classes and their distinct queues, separate RED instances for each class can be independently configured, with lower drop thresholds for traffic in lower priority classes. In addition, different drop profiles can be applied to traffic within a queue based on whether the traffic is within committed service parameters (green), within peak service parameters (yellow), or out of profile (red). The result is called Weighted RED (WRED) because drop probability is weighted to favor high priority traffic classes and traffic within each class that conforms to any service parameters specified.

For example, with QoS configured, long-term TCP flows can be assigned to a separate traffic class with a large buffer allocation and large share of the link bandwidth. In the context of the earlier discussion of the optimum buffer size for long-term TCP sessions, the RED min threshold for this traffic class could be set equal to the RTT multiplied by the minimum bandwidth allocated to long-term TCP traffic class.

\[ \text{Min Threshold} = \text{RTT} \times \text{BWtcp\_long\_term} \]

The max threshold could be set close to RTT \(\times\) BWlink, which would be the full buffer capacity of the port. If VoIP traffic is assigned to a strict priority queue, this will require only a small share of the egress buffer capacity and bandwidth.

### Optimum Buffer Architectures for Matrix Switches

For high performance switches designed to support numerous 10 GbE ports, the most common switch architecture is based on a cross point matrix switch fabric. With these architectures, it is necessary to buffer traffic on both the ingress port and the egress port. Ingress buffering is required because there is non-zero probability of momentary congestion within the switch fabric itself. Egress buffering is needed because the egress port may be congested by aggregated traffic from multiple ingress ports. Because of its high bandwidth, fabric congestion is typically very short lived, which means that ingress buffers can be much smaller than egress buffers.

One of the most important functions of the ingress buffer architecture for matrix switches is to eliminate the possibility of Head-of-Line (HOL) blocking. HOL blocking occurs when the packet at the head of the queue cannot be forwarded immediately because some other ingress port is transmitting traffic over the fabric interface of the destination egress port. Thus, with FIFO queue scheduling, packets deeper in the ingress queue are blocked in spite of the fact that they may be destined for idle egress points from the fabric.

HOL blocking is eliminated by using a Virtual Output Queuing (VOQ) discipline instead of FIFO to schedule access to the switch fabric. With VOQ, the scheduler looks beyond the head of the queue to find the next packet that is destined for an unoccupied fabric egress interface and forwards that packet before the head-of-line packet.

VOQ is optimized when the packet processing ASICs are fast enough to perform the lookup of the destination port before the packet enters the ingress buffer. This allows VOQ to immediately forward any packet in the queue to the correct output port that isn’t currently occupied. Suboptimal implementations may enqueue the incoming packet after reading only the source address and before the destination lookup has been completed. In this event, a VOQ implementation may assume that the packet is destined to the same output port as the last packet from this source. If this assumption is wrong (i.e., when a source is simultaneously forwarding to multiple destinations), the result is either wasted fabric bandwidth or wasted buffer space and unnecessary delay.

An additional enhancement to ingress buffering implementations is to apply ingress QoS policies such as classification, traffic policing, packet reordering, and congestion avoidance before the incoming packet reaches the VOQ. This ensures that neither out-of-profile packets nor packets that are destined to be randomly dropped can waste ingress buffer capacity. Ingress QoS also ensures that any packet loss that does occur at the ingress buffer has minimal effect on high priority traffic. If the QoS function is applied only on the output port,
any ingress drops will affect all traffic classes equally and time-sensitive traffic will experience more latency and jitter.

For switch/routers with high densities of GbE and 10 GbE ports, each line card may support a number of ports. If separate buffers are dedicated to each port, the cost of buffering scales linearly with the port capacity of the switch. The overall cost of buffering can be very significant, especially if the buffers are scaled to accommodate high RTTs of MAN/WAN links.

Another approach that provides the needed buffer capacity at significantly lower cost is to allow the multiple ports on a line card to share a common pool of buffer capacity. Each port on the card can then receive a dynamic allocation of buffer space based on the traffic load. A shared pool of buffers is much more efficient than dedicated buffers because no buffer capacity is wasted on idle or lightly loaded ports.

Buffering and Traffic Management in the Force10 E-Series Switch/Router

The Force10 E-Series switch/routers feature a VOQ matrix switch architecture with robust ingress/egress buffering (up to 200 ms on 10 GbE interfaces). For line cards with multiple ports, the buffers are implemented as a shared pool of memory resources.

The ASIC-based forwarding plane delivers line-rate forwarding at 10 Gbps and beyond with any combination of ACL, WRED, QoS, and other advanced features or services enabled. Traditional architectures rely on additional sequential processing to implement each service that is configured. Sequential processing at less than line-rate increases packet latency and degrades overall forwarding performance, leading to unpredictable system behavior.

Full line-rate packet processing allows the E-Series buffer management functions to be optimized for both the ingress and egress buffers and ensures the most efficient utilization of the Terabit-per-second switch fabric.

Figure 4 shows how packets are forwarded through the E-Series switch/routers. The packet-forwarding ASICs illustrated work together to implement all supported services as described in figure 4.

The Packet Processing Engine (PPE) performs the Ethernet MAC level functions. The PPE passes (2) a condensed version of the header information to the Packet Classifying Engine (PCE). The PCE performs simultaneous line-rate Ternary CAM (TCAM) lookups (3) in order to classify the packet based on the Layer 2 (MAC) or the IP 5-tuple. In addition to looking up the destination port, the TCAM returns information that specifies how the packet should be handled to conform to any configured ACLs or QoS policies. At this point, the PCE also identifies and separates routing and control messages from data packets, thereby ensuring that control traffic is not delayed by data traffic flowing through the switch. The PCE can retrieve all the information on how to deal with the packet in the time it takes to receive just the packet header. The E-Series architecture supports TCAMs with the capacity to handle up to hundreds of thousands of TCAM entries.
When the table lookups are complete, the PCE passes information (4) back to the PPE specifying modifications to the Ethernet or IP packet header (e.g., to mark the packet with its appropriate traffic class using CoS, DSCP, TOS, or MPLS EXP bits) and populating an internal header appended to the packet that includes the destination egress port and other internal packet handling information. When these packet modifications are complete, the packet is passed (5) to the ingress Buffer and Traffic Manager (iBTM).

The iBTM enforces the configured policies that were identified during the packet classification process. Supported policies include: two-rate, three color token bucket rate policing, RED or WRED for congestion avoidance, ingress buffer management to support multiple QoS queues, and queue scheduling using FIFO, Strict Priority, or Weighted Fair Queuing as required. The combination of token policing on the ingress and WFQ on the output queue means that it is possible to specify guaranteed upper bounds on queuing delay. After the iBTM processing is completed, the packet is passed to the Switch Fabric Scheduler (SFS), which implements VOQ in scheduling the ingress port’s access to the Terabit Switch Fabric (TSF) ASICs (6).

On the egress side, the egress BTM (eBTM) receives the packet from the switching fabric (7), and enforces egress policies. The eBTM supports the same basic functions as the iBTM: token bucket metering (which now can be used for rate limiting), congestion avoidance with RED/WRED, egress buffer management, queue scheduling, enforcing ACLs, etc. However, in this case the policies are being applied to egress traffic aggregated from all the active ingress ports. Finally, the packet is passed to the egress PPE (8). The egress PPE removes internal header information, modifies the MAC, and directs the packet out the appropriate port (9).

**Conclusion**

Although selecting the optimum port buffer size for switch/routers is still a rather inexact science, there is general consensus that buffers scaled to the delay-bandwidth product are the best choice for maximizing link utilization for long-term TCP applications. Therefore, ports on switch/routers need buffers that can be scaled to accommodate the maximum expected end-to-end propagation delay, often the largest component of RTT.

Large buffers shared by all applications in a FIFO queue can present problems for delay-sensitive applications such as VoIP. However, QoS functionality provides an effective way to prevent these applications from experiencing excessive queuing delays, irrespective of the overall buffer size.

Used alone, or in conjunction with QoS, congestion avoidance mechanisms based on RED can further improve link utilization by reducing the probability of global synchronization caused by tail drop.

Force10 E-Series switch/routers are based on a line-rate packet forwarding plane and cross point matrix architecture that provides optimum utilization of large port buffers.

- GbE and 10 GbE ports have 55 to 200 milliseconds of buffering. Buffer management leverages cost-effective, dynamic access to a pool of buffers shared by all the ports on the line card
- The combination of line-rate packet processing in the forwarding plane and ingress QoS ensures efficient usage of ingress and egress buffer space and optimized VOQ functionality.
- Line-rate WFQ at 10 Gbps is a unique QoS capability of the E-Series ASICs. WFQ, in conjunction with token bucket policing, can be used to deliver a guaranteed upper bound on queuing delay over an end-to-end network of E-Series switch/routers.