Enhanced Transmission Selection – IEEE 802.1Qaz

Victor Lama
Dell Fabric Specialist G500
23 March 2011

What You Will Learn

You will gain an understanding of the specifics regarding the Enhanced Transmission Selection standard (ETS), as defined by the IEEE committee. You will also be presented with some background information regarding established QoS mechanisms, which will help provide a contextualized and holistic view of how transmission selection has been addressed in the past. You will observe that ETS is the next logical step in the progression of data forwarding technology.

Overview

Converged Ethernet carries multiple traffic types that are sensitive to different aspects of data transmission. For example, storage traffic is sensitive to packet loss, while IPC traffic is latency-sensitive. In a single converged link, all these traffic types need to coexist without imposing serious restrictions on each other's performance.

The differing service needs of applications supported on a consolidated Ethernet are represented by separate traffic classes. These applications can be bursty in nature as they transfer large files. Therefore, support for these classes on the same links requires the ability to allocate a guaranteed share of bandwidth to each of them, while also allowing each class to exceed that minimum guaranteed bandwidth if another class is not fully utilizing its share. This is where the Enhanced Transmission Selection algorithm is applicable.

In February of 2011, the IEEE 802.1 committee completed draft 2.4 of the Enhanced Transmission Selection standard. ETS is one of the four Data Center Bridging (Converged Enhanced Ethernet) standards that have been ratified to make unified fabric networks possible. The standard defines a framework for allowing multiple classes of network traffic that traverse the same link to be guaranteed a minimum amount of the link's available bandwidth. ETS exploits the time periods in which the offered load of a particular Traffic Class (TC) is less than its minimum allocated bandwidth by allowing the difference to be available to other traffic classes.

> NOTE – The phrase "Traffic Class" (upper case) refers to a defined traffic type as per ETS semantics. The phrase "traffic class" (lower case) can either be a generic reference to a traffic type or traffic that has been classified using 802.1p Class of Service semantics.

The ETS standard does not preempt or necessitate the removal of any queue mapping and scheduling primitives that already exist in the forwarding paradigms of 802.1Q bridges. ETS can be categorized as more of an overlay that specifically addresses scheduling, but only insofar as it pertains to "policing" the bandwidth

allocation requirements for each configured TC. Traffic classification and enqueuing are relegated to the configured QoS mechanism deployed on the device.

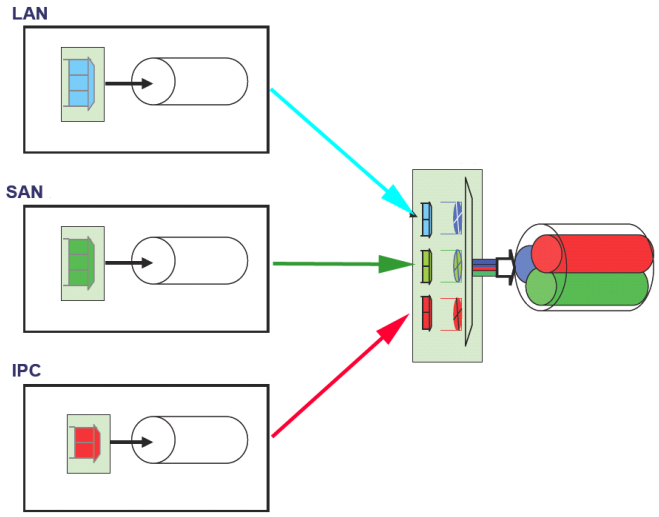

Figure 1 Convergence of Ethernet

ETS and the Evolution of Existing QoS Mechanisms

As its name suggests, ETS is really an enhancement to existing transmission selection mechanisms, therefore, a brief review will prove useful in understanding its evolution and the problem it solves.

Transmission selection refers to the decision-making process that an interface packet scheduler makes when determining which packet needs to be forwarded next. When all the traffic traversing the link has similar forwarding characteristics and delivery requirements, the transmission selection process is rather straightforward. In such situations, all data frames are given the same attention and allocated the same amount of I/O resources. The challenge then is one of serializing the frames as quickly as possible as they arrive at the interface's hardware queue.

During periods in which a particular transmitting interface is not able to forward the frames as fast (or faster) than the rate at which they are being received (for example, if the output interface is oversubscribed, aka Head-of-Line Blocking), it becomes necessary to utilize software queues to store the data while the hardware queue is given time to empty itself, and then accept the cached frames and serialize them onto the output link.

The software queuing architecture is enabled when a QoS mechanism is configured on the device; otherwise, the only output queue that exists is the hardware queue. Regardless of the queue type, incoming frames will be dropped by the output queue when its associated buffers become full. This is known as tail-drop. The tail drops force TCP retransmissions, which result in the data-transfer rate slowing down, thereby causing the link's bandwidth utilization to decrease. This situation is an example of *congestion management*.

NOTE – Virtual Output Queues (VOQ) are input buffers that are mapped to the output buffers and are used to alleviate the problem of Head-of-Line Blocking. Vendors have the choice of deciding just how sophisticated the architecture will be when mapping VOQs to output queues.

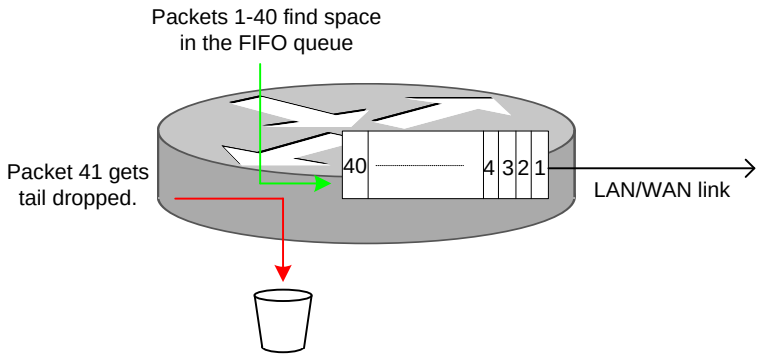

Figure 2 FIFO Queue and Tail Drop

There are QoS mechanisms that deliberately drop packets in anticipation of the queue becoming full, such as Weighted Random Early Detection (WRED). This is a *congestion avoidance* mechanism and it represents a preemptive approach to addressing network congestion. The recently adopted IEEE DCB standard known as Per-Priority Flow Control is also an avoidance mechanism that tracks buffer utilization – only it does not act to prevent congestion, but only to protect a lossless class of traffic, such as FCoE, from losing packets.

In its simplest form, the scheduling algorithm will forward each packet in the order in which it was received. This is known as FIFO queuing – first-in, first-out. By default, the hardware queue always uses FIFO scheduling, whereas the manner in which frames are selected for transmission among the software queues depends on the scheduling algorithm configured on the device. Parenthetically, the queues do not actually store the frames, the buffers do. The queues only contain pointers that point to the buffers in which the packets reside.

FIFO queuing can prove to be sufficient when the offered workload is relatively mild and/or when the traffic characteristics are somewhat homogenous. The practice of deploying more complex QoS mechanisms, and thereby offering more discriminating forwarding services, gained prominence with the advent of the first iteration of what we refer to as converged networking; namely, when voice traffic was packetized and placed on the same network as application data (Voice Over IP - VoIP).

Unlike email and web traffic, digitized voice packets and control signaling are extremely intolerant of delay and jitter (variable delay) and must be given expedited processing when it comes to buffering and forwarding. In such cases, there is no congestion or even anticipated congestion, but simply a recognition that there are different traffic types traversing the link, each with its own forwarding requirements. This situation can be described as *bandwidth management* or *queue management*.

Queue Management and its Relation to Bandwidth Usage

Some of the QoS scheduling mechanisms that have been used on layer 3 networking devices in the past are Priority Queuing (PQ), Custom Queuing (CQ), Fair Queuing (FQ), Weighted Fair Queuing (WFQ), Class-based Weighted Fair Queuing (CBWFQ) and Low Latency Queuing (LLQ). CBWFQ and LLQ are Cisco proprietary. All of these approaches define in one form or another the manner in which packets are classified, queued (placed in buffers), and scheduled for transmission.

The greatest differentiator among all them is the manner in which the scheduler selects packets for transmission; more specifically, the relative importance it assigns to certain queues as it services them. Note that the rate at which frames are extracted from a queue is directly proportional to the amount of bandwidth that the particular traffic class will enjoy.

A commonly used approach for assigning relative importance to multiple traffic types is to attach a "weight" – or a quantified expression of importance – to each queue. The assigned weight translates into a definitive number of frames that will be selected for transmission as each queue is serviced. Depending on whether a strict priority queue is utilized – in which case it will always be serviced as long as a packet resides in its associated buffers – all the remaining queues are serviced in a round-robin fashion.

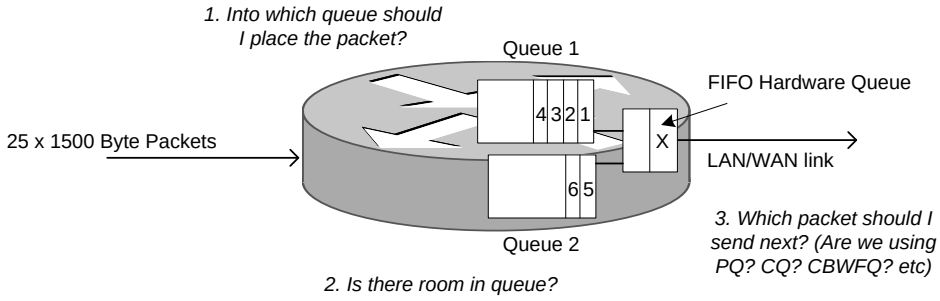

Figure 3 QoS Enabled. Multiple SW Queues, 1 HW FIFO Queue

On layer 2 devices, the QoS mechanism of choice is known as Weighted Round-Robin (WRR), which may include a priority queue. This is similar to the mechanisms used in routers. The combination of applying weights and servicing priority and standard queues has the equivalent effect of assigning certain percentages of link bandwidth to the traffic classes that reside in the serviced queues.

Stated differently, WRR allocates bandwidth to queues in proportion to their weights. For example, a layer 2 switch's interface queues may be configured (weighted) such that, upon each round of queue servicing, the following transmission selection occurs: 50 packets selected from queue 4, 25 from queue 3, 15 from queue 2, and 10 from queue 1. This equates to allotting 50% of the link bandwidth to queue 4, 25% to queue 3, 15% to queue 2, and 10% to queue 1.

The Enhancement to Transmission Selection

The QoS mechanisms discussed thus far have static weight assignments that the interface scheduler acts upon in a predictable fashion. Each traffic class will be queued and then scheduled for transmission in a manner defined by the relative weight assigned to them. Concomitantly, the bandwidth allocated to each traffic class is also static. The underlying assumption of such an approach is that the different traffic classes will offer similar loads for transmission at somewhat predictable rates. This is not the case with today's converged networks, which leverage a "fat pipe" to carry various types of LAN, storage, and other types of workloads.

The IEEE 802.1Qaz standard provides an operational model and associated set of requirements that allow for *dynamic* bandwidth allocation to multiple traffic classes as the need presents itself. It does not, however, provide specifics on the kind of scheduling algorithm to use or other implementation details, which leaves some room for misunderstanding and debate.

Note, the technique of dynamic bandwidth allocation is not new and has its roots in what is known as Statistical Time Division Multiplexing (STDM).

What's Behind ETS's Dynamic Bandwidth Allocation?

A review of how TDM works will give some insight into how ETS handles varying rates of offered traffic.

In figure 4 below, each DS0 has a bit rate of 64,000 bps and represents a digitized voice channel. The aggregate bit rate of all 24 channels, plus overhead, is 1.544 Mbps (DS1). To achieve these rates, the multiplexer uses a round-robin scheduler that takes an 8-bit sample from each DS0 and places the bits on the output channel. One round of scheduling results in 192 bits (8 bits from each of the 24 channels, starting with channel 1 and proceeding in order to the 24^{th} channel). Added to the 192 bits of a single round of scheduling is a framing bit, for a total of 193 bits. Therefore, in one second, 8,000 rounds (frames) of sampling are performed, yielding 64,000 bits of offered traffic from each DS0 and an aggregated rate of 1.544 Mbps.

On the receiving end, the 1.544 Mbps data stream is de-multiplexed in a deterministic fashion: one channel at a time (1 to 24), 8 bits per sample, 8,000 samples per second. It's the exact reverse of what occurred on the sending end. The specific position of the 8 bits in the stream is how the multiplexer at the receiving end knows to which channel the 8 bit samples belong.

The scheduler at the receiving end expects to sample 8 bits from each channel, no matter what. Therefore, each channel *must* produce an 8 bit sample for each round of scheduling. If a channel is constantly in use, there is no issue. But what happens when a channel is idle because the phone or FAX machine or modem is not in use? In other words, what if a particular source of traffic is not offering its minimum allowable traffic rate? In that case, 8-bit keepalive signals must be generated to satisfy the fixed position, time division scheme.

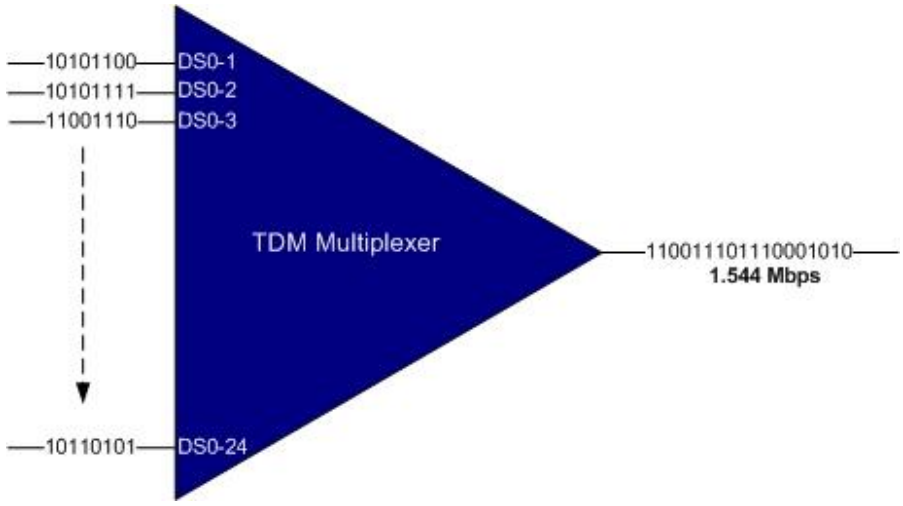

Figure 4 TDM Multiplexer

Imagine that channels 13 through 24 are inactive at any given time; that results in 768 Kbps of bandwidth being utilized for no other purpose but to send keepalive signals! To remedy this situation, a statistical multiplexer is oversubscribed with more than 24 DS0 channels (perhaps 32), allowing the bandwidth of an idle channel to be re-allocated to another channel that has an actual need to send data. It is for this reason that statistical multiplexers are sometimes referred to as bandwidth managers. (I can remember configuring the IDNX-20 STDM back in 1994)

With ETS, different traffic classes, as defined by their 802.1p CoS markings, are placed in a numbered Traffic Class. Furthermore, each Traffic Class (TC) is serviced by a certain scheduling algorithm. An ETS TC (that is, a queue or series of queues that are slated for the ETS scheduling algorithm) is configured for each major type of traffic (LAN, SAN, etc) and is allocated a minimum guaranteed percentage of available bandwidth. ETS then leverages a statistical multiplexing algorithm to dynamically re-assign bandwidth to the different Traffic Classes as their offered rates increase or decrease. ETS classification semantics are further detailed in the "Understanding ETS Traffic Classification and Scheduling" section below.

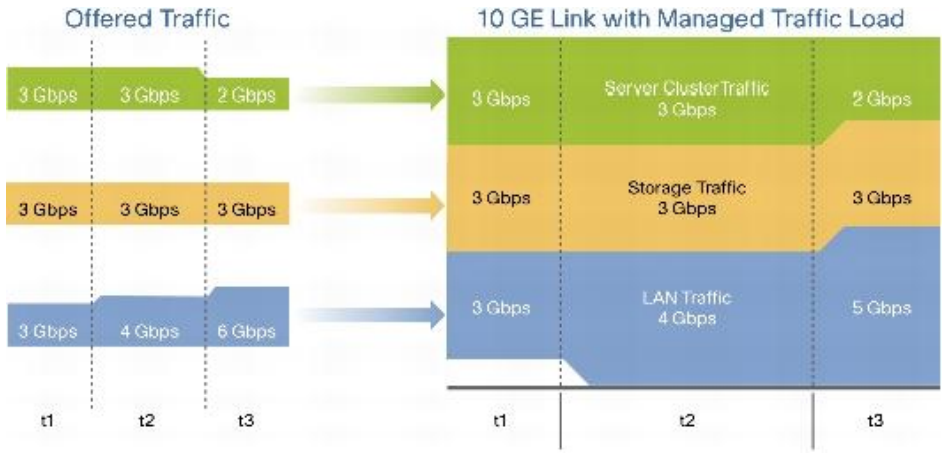

Figure 5 Dynamic Bandwidth Allocation using ETS

ETS Compliance Standards

The ETS standard places the following compliance requirements on VLAN-aware bridge vendors:

1. Must support the ETS Scheduling Algorithm
2. Must support DCBX

For a VLAN-aware bridge to be in conformance with the ETS Scheduling Algorithm itself, the following primitives must be supported:

1. Support for at least 3 TCs (TC);
   NOTE — a minimum of 3 traffic classes allows a minimum configuration such that one traffic class contains priorities with PFC enabled, one traffic class contains priorities with PFC disabled, and one traffic class using strict priority.

2. Support bandwidth configuration with a granularity of 1% or finer
3. Support bandwidth allocation with a precision of 10%;
   NOTE – this means that each configured bandwidth percentage must be met with an error margin of no more than +/- 10%.

4. Support a transmission selection policy such that if one of the traffic classes does not consume its allocated bandwidth, then any unused bandwidth is available to other traffic classes
5. Support DCBX

Understanding ETS Traffic Classification and Scheduling

One of the most confusing aspects of the ETS standard is the taxonomy, which is in constant flux within the IEEE committee itself. The lexicon used to define traffic classification has been revised and redefined several times over the course of the last 2 years. As of the latest iteration, v2.4, the following semantics are defined:

- Application traffic is prioritized using 802.1p semantics. Eight (8) priorities are defined (0-7).

- Each ETS TC must allow the inclusion of several priorities of traffic. ETS uses a 4-bit field to quantify the maximum number of TCs that can exist. This means 16 TCs can be defined (0-14, 15 has been excluded), with a minimum of 3 required for compliance purposes.

- A Transmission Scheduling Algorithm (TSA) is used for each TC and is represented by an 8-bit field in the ETS TLV of the DCBX protocol. That means up to 256 TSAs can be applied to the different TCs defined (0-255, where 0 is strict priority, 1 is credit-based shaping, 2 is ETS, 3-254 are reserved for future use, and 255 is vendor specific). ETS specifically addresses 3 scheduling algorithms – 0, 1 and 2.

As just stated, ETS specifically recognizes 3 types of Transmission Selection Algorithms (with a theoretical maximum of 256) that can be applied to different workloads, as well as a QoS order of operations.

First, there are those traffic types that require strict priority queuing, such as VoIP and IPC. Depending on the vendor's implementation, the strict priority queue will be serviced as long as a packet exists in its queue,

which can result in starving the other queues. The Cisco LLQ algorithm remedies this situation by policing the strict priority queue. Regardless of the vendor's approach, within the framework of ETS, strict priority queuing is recognized as a requirement for converged links. Therefore, queues configured for strict priority queuing will be serviced first.

The second type of scheduling algorithm would service workloads from Audio/Visual applications, such as real-time IPTV. Such workloads would be placed in queues configured for another developing IEEE standard referred to as IEEE 802.1Qav, Forwarding and Queuing of Time-Sensitive Streams using a credit-based shaping algorithm. It is one of the 4 protocols that fall under the designation of AVB, which is being ratified by the IEEE Audio Video Bridging Task Group. In short, it leverages a form of end-to-end bandwidth reservation for each AV traffic flow, similar to the RSVP approach from the QoS days of yore.

Lastly, there are those workloads that require dynamic bandwidth allocation to ensure a lossless transmission path, such as FCoE or iSCSI traffic. IP-based mission critical and best-effort based delivery applications would also fall under this category. These workloads will reap the rewards of ETS's innovative approach to dynamic bandwidth allocation, thereby ensuring their flows are given the resources they need for successful transmission and reception. The queues configured for ETS scheduling will be serviced last. In other words, the bandwidth that ETS dynamically allocates is the bandwidth that remains *after* the strict priority and credit-based shaping algorithms have serviced their queues.

To ensure that each bridge in the data path is informed of the configured ETS classification and scheduling scheme, the DCBX protocol will exchange configuration TLVs upon link initialization.