Choosing the Right TSN Tools to meet a Bounded Latency

By: Don Pannell, Fellow, NXP Semiconductors
donald.pannell@nxp.com

Moving from Domain to Zonal

As Automotive moves from Domain to Zonal networks, many kinds of data end up sharing the same wires. IEEE 802.1's Time Sensitive Networking (TSN)[1] standards are the solution.

TSN may be new to many people even though the IEEE 802.1 Work Group has completed more than a dozen TSN standards over the last decade & many are already supported in HW available today. Many of these TSN standards are targeted at bounding the latency of a steam. With so many standards or "tools" available, the challenge of knowing what the proper "tool" to use and when, is increasingly harder. This is especially true when considering that the TSN tools are intended to work together on the same port at the same time!

[1] See https://1.ieee802.org/tsn/

Introduction

This article focuses on the completed TSN standards (at the time) that control latency. An overview of each "tool" is given so the reason the standard was created and the target problem each tool was designed to solve is known. The relative "costs" of using each tool (in terms of silicon complexity, of network configuration complexity, and in network efficiency) is examined and compared. Based on these "costs", a per-hop metric is proposed so a network engineer can determine which tool should be used and when. A proposed hierarchy, or priority order of tool usage, is shown.

The Shaper Standards – In the Order of their Development

The Strict Priority Shaper – 802.1p-1998

The earliest IEEE 802.1 shaper was standardized in 1998, more than two decades ago! It was documented in the IEEE 802.1p-1998 project and it defines the Strict Priority Selection algorithm used to select the next frame to be transmitted out a port when a port supports more than one priority queue. These priority queues are referred to as Traffic Class (TC) queues in the standard. Today, there are many supported mechanisms in 802.1 that can be used to determine which Traffic Class a frame get enqueued to. While most of these mechanisms are beyond the scope of this article, the original mechanism of using the per-port mapping of the frame's 3-bit Priority Code Point (PCP) field contained in 802.1 Tagged frames is an integral part of TSN – as this is the mechanism used to indicate that some frames need to be treated differently. This will be seen more later.

Originally, the Strict Priority mechanism was created to solve real-world problems of the time, one of which was effective network management. Simply put, a network can't be managed using the same wires that "data" traffic uses, if the "data" traffic is using all of the wire's bandwidth and there is no preference (i.e., a higher priority) given to the network management traffic. While this situation appears to be unlikely, it is unfortunately quite easy to create. All it takes is a bridge loop – i.e., two ports connected on the same bridge or more than one connection between two bridges (on bridges that don't support link-aggregation). Visualize a network connection (wiring) closet in a building with 100's of wires and you can see it was quite easy to accidently plug a cable into the wrong jack, creating a bridge loop, bringing down the entire network. I have seen this situation persist for days until the looping wire was discovered and disconnected.

IEEE 802.1 Spanning Tree solves this problem if bridges place the Network Management traffic (i.e., the Spanning Tree protocol frames in this case) in a higher priority Traffic Class queue from the other "data" traffic. This allows the Network Management traffic to get through from bridge to bridge, so Spanning Tree can detect and automatically (logically not physically) disconnect the bridge loop.

Figure 1, below, shows an egress queue structure typically used in the transmit path of bridges (the IEEE 802.1 name for what most people call switches). This same structure can be used in end stations NICs

(Network Interface Controller) as well. The 1998 standard supports 1 to 8 Traffic Class queues and the figure shows the default mapping of a frame's 3-bit PCP value to Traffic Class queue for an 8 queue design (PCP 0 being a higher priority than PCP 1 is as it is in the standard). The highest Traffic Class (8) is labeled Network Management as this is the queue where these frames need to go to solve the bridge loop problem (and other network management issues too).

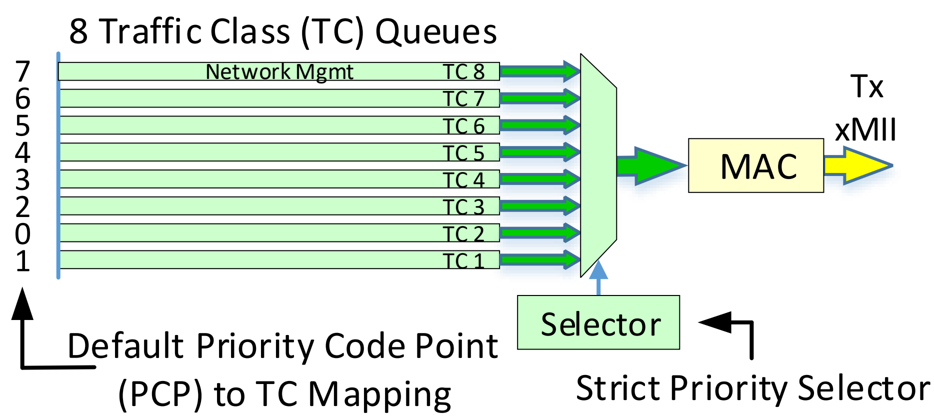

Figure 1: Basic Egress Queue Model with Strict Priority

The 1998 standard supported two Traffic Class draining algorithms, Strict & Proprietary. The Strict algorithm is defined as: "Frames are selected from the corresponding queue for transmission only if all queues corresponding to numerically higher values of traffic class … are empty at the time of selection."

If the Strict Priority Selection algorithm is selected and the PCP to TC mapping table insures only network management frames are placed into the highest queue, then Network Management frames ***will*** get through so they can be processed.

Note: Network Management frames are usually not Tagged, so their PCP value needs to be assigned by other mechanisms. Refer to each respective IEEE 802.1 standard for more information.

The Credit Based Shaper – 802.1Qav-2009

More than 10 years passed before a new shaper was standardized in IEEE 802.1. That new shaper, the Credit Based Shaper (CBS), was targeted to solve issues associated with audio and video streams, but it can also be used for other data types as well. Why? Because the Credit Based Shaper was designed to solve the generic problem of bursting, which occurs regardless of the data type (audio, video, or "data"). Bursting (the transmission of multiple back-to-back frames) occurs naturally in bridges on bridge ports that are congestion points (discussed below) and in end-stations because of the way end-station (NIC) hardware & software is designed. NICs are optimized to get the highest data throughput possible and use bursting to attain this goal. While this approach gives the best performance for the end-station, large bursts of frames is the worst possible traffic in a network (i.e., for bridges).

Bursts of traffic from a single source is not a problem, just as one company's employees all going home at the same time does not affect the city roads. But bursts from multiple sources where there is even a single network congestion point, is a problem. Think of multiple company's employees all going home at the same time. The city roads can't handle it, and latency (the time to get home) goes up.

If the network congestion point is in a bridge, that bridge could drop packets. Examples of congestion points are a Gigabit port switching data to a 100 Mbit port, or two 100 Mbit ports switching data to a single 100 Mbit port. In both cases the output data rate is less than the potential input data rate. Ethernet bridges have been successful working this way for decades as higher layer protocols re-transmit the missing data. But with today's newer applications that require low latency, along with guaranteed delivery of the network data, the re-transmission of dropped packets is not an option.

Adding memory to a bridge to minimize packet drops during bursts is not an option either. While this could solve the problem in a specific network configuration, it doesn't really solve the problem generically. For example, if another port's flow is added to the already congested flows, the added memory may no longer be enough. Adding memory also increases latency which goes against the goal of low latency. If bridges were required to support 2 seconds of buffering, then a three-hop bridge network would contribute up to 6 seconds of added latency. At this point re-transmitting the dropped data frames is likely faster.

The Credit Based Shaper (CBS) addresses the packet dropping problem at the network congestion points & NICs by de-bursting the flows in hardware. It is configurable on a per Traffic Class basis, so flows mapped to a Traffic Class queue where CBS is disabled, perform as they normally would. And flows mapped to a queue where CBS is enabled automatically have their transmit bursts removed without any changes to the "optimized" NIC drivers (nor the bridge's enqueueing hardware).

The Credit Based Shaper:

- Limits the bandwidth a Traffic Class queue can transmit (in hardware)
- De-burst flows (in hardware) by delaying the transmission of each subsequent frame
- It allows for very small bursts of data to 'catch-up' (by the accumulation of credit due to momentary interference) so the Reserved data rate can be maintained
- In bridges only one CBS is needed per Traffic Class to de-burst the traffic at each hop
- In end-stations one CBS is needed per flow followed by one per Traffic Class

The Strict Priority Scheduler (discussed above) is a necessary and integral mechanism in making the Credit Based Shaper work in a functioning system. Figure 2, below, shows where the Credit Based Shapers reside (along with the other shapers that will be discussed later). In this example it is important to note that the Network Management flows are moved to Traffic Class 6 and CBS Class A (PCP = 3) & Class B (PCP = 2) are in Traffic Class queues above Network Management. This works because the Audio Video Bridging (AVB) profile specification (IEEE 802.1BA) limits the total transmission rate of all the Class A + Class B flows to be no more than 75% of the port's output data rate. With Network Management being the top priority of the remaining (non-reserved) bandwidth ensures that it can perform its original function (as IEEE 802.1 Network Management protocols are designed to use very little bandwidth).

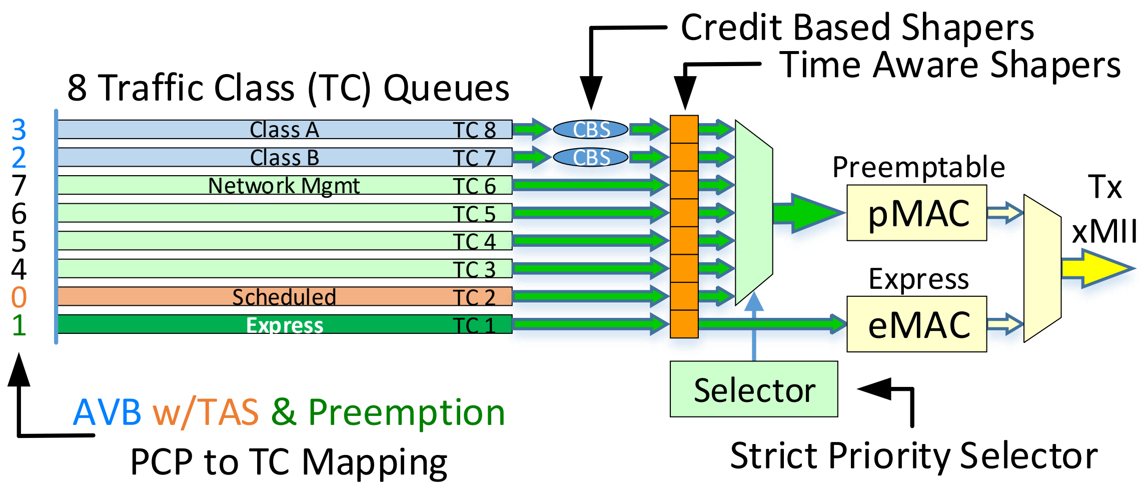

Figure 2: Example Simultaneous TSN Shaper Egress Queue Model

Note: The AVB profile specification (IEEE 802.1BA) defines specific parameters & requirements that affect the Credit Based Shapers & the resulting latencies for Class A & Class B. This was necessary in the AVB profile as it satisfies a plug-and-play use case. The shapers discussed below assume their networks are engineered (not plug-and-play). The Credit Based Shaper can also be used in these engineered environments, using the same or very different parameters & requirements from what is specified in the AVB profile.

The Time Aware Shaper – 802.1Qbv-2015

After the Audio Video Bridging (AVB) set of standards were completed for the plug-and-play audio/video use case (in 2011), it was clear that there were use cases that required lower latency from what AVB could provide. And these implementers were willing to engineer the network if needed (i.e., limit the supported topologies, etc.) to get these lower latencies.

One approach to achieve lower latencies, is to find a target use case and solve its latency need. But if a new use case emerges that needs an even lower latency, the previous solution may not work. An alternate approach is to find a solution that achieves the theoretical lowest possible latency. This is exactly what was done by the author and the Time Aware Shaper (TAS) is what was proposed[2] and standardized (IEEE 802.1Qbv).

The Time Aware Shaper works best in applications that are cyclical, i.e., there are fixed periodic intervals when critical sensor data is transmitted & processed, and the resulting critical actuator commands are issued. Most industrial machines work on this concept as do many other systems.

---

[2] http://www.ieee802.org/1/files/public/docs2011/new-pannell-latency-options-0311-v1.pdf.

TAS minimizes the latency for time critical flows by ensuring that the port a frame is egressing (bridge or end-station) is idle at the time the critical flow is scheduled to egress the port. This minimizes the latency since the critical flow can start transmitting precisely at its scheduled time, as the port is not busy transmitting anything else (like a maximum size interfering frame). The timing of the packet transmission is accomplished by the addition of transmission gates to the output of each Traffic Class queue (the orange boxes in Figure 2, above). A timing schedule of gate opening and closings for each Traffic Class is also needed. If a gate is closed on a queue that TC cannot transmit any data. When TAS is not being used all the gates on all the queues are open all the time.

For TAS to work, the transmitting devices in the network need to be "time aware", i.e., they need to be "in-sync" with each other by running some form of the Precise Timing Protocol (like IEEE 802.1AS or IEEE 1588). And the hardest part, a schedule for the time critical flows needs to be developed and loaded into each of the network devices.

The schedule becomes easier to solve by minimizing the number of critical flows & Traffic Class queues that must use this extremely lowest latency mechanism & by minimizing the number of gate opens & closes per cycle (i.e., the periodic interval). This minimalization is shown in Figure 2, above. Only one TC is shown as "Scheduled" and this is TC 2 indicated by a Priority Code Point (PCP) value of 0. How can a very low priority queue get lower latency than the TC queues above it? Easy, if it is the only TC with a gate open it is the highest priority during its window of transmission.

Note: To ensure an egress port is idle at a specific "Scheduled" gate open time, the other queues should not start transmitting a new frame if that frame cannot finish before its gate closes. While there are various mechanisms to insure this, the net result is a loss of network utilization of the port for a period of time with each "Scheduled" gate opening. This "cost" is part of the shaper "tool" comparison below.

Preemption – 802.1Qbu-2016 & 802.3br-2016

Preemption was developed as a standard at the same time as the Time Aware Shaper, discussed above. It was an alternative for networks that were not Time Aware, and it too gets very low latency, but not as low as the Time Aware Shaper. Preemption is easier to use as there are no "Schedules" to work out. But it cost a lot in hardware, only works if its link partner also supports preemption, works best at 100 Mb/s or slower links, and the standard supports only one level of preemption, all of which minimizes its potential use cases.

Preemption does what it says. It interrupts a lower importance "preemptable" frame in the middle of its transmission. It then allows the higher importance "express" frame or frames to egress the port. Only then does it resume transmission of the "preempted" frame where it left off (with appropriate headers) as it does not re-start transmission back at the beginning of the frame. The "preemptable" frame can be interrupted in this way multiple times.

IEEE 802.1 could not provide the complete standard for preemption on its own as the mechanism described above had to be done in the Ethernet MACs, which are the responsibility of IEEE 802.3. The preemption MAC mechanism & the codes sent down the wire is standardized in IEEE 802.3br. It defines 2 MACs, a pMAC (for Preemptable MAC) and an eMAC (for Express MAC). These can be seen on the right-hand side of Figure 2, above. Preemption on Ethernet has the following restrictions:

- Minimum fragment size is 64 bytes (like any standard minimum Ethernet frame)
- Fragment padding up to 64 bytes is not done so a fragment could be nearly twice the fragment size
- Default fragment size is 64 bytes, but could be negotiated by the link partner to be 128 or 256 (certain applications may need to see more of the header(s) for proper operation)

The IEEE 802.1 standard for preemption (IEEE 802.1Qbu) defines which Traffic Class queue or queues are connected to the two MAC types. If more than one TC queue is connected to a MAC, the connection is through a Strict Priority Selector. In Figure 2, Traffic Class 1 is the only Traffic Class connected to the eMAC. That means that any frame enqueued to TC 1 will preempt all the other Traffic Class queues (i.e., any data that is currently transmitting). So even though TC 1 is the lowest priority queue, the way it is connected, it is effectively above all the other queues since it can preempt them!

Note: The Preemption Traffic Class, TC1 in Figure 2, is the highest priority only if its Time Aware Shaper gate is open.

The Metrics for the TSN Shaper Standards

The IEEE 802.1 Working Group designs their standards as tools that can be used together, if possible. This is not an easy task with the multiple decades of standards to consider. For the TSN Shaper Standards discussed above, this means that the various shapers can work together at the same time, as each Traffic Class queue on a port can be configured with a different shaper (assuming the device you are using supports this). An example of this is shown in Figure 2, above.

This flexibility makes the job of the network designer/engineer harder unless the “specs” on the supported tools are known. There are many ways to measure or evaluate a tool. The ones mainly focused on here are cost & performance. Availability is also an important metric as the perfect tool can’t help you if you can’t get it at any price. In general, the newer IEEE standards are less available.

The TSN Shaper Cost Guidelines

Table 1 shows selected costs that are examined in this comparison of the TSN Shapers. Which are:

- Silicon Complexity: The amount of die area a shaper requires. This likely affects the device cost & availability of the shaper, & it clearly affects the engineering time to build it. This is important as no silicon maker wants to spend money making an expensive feature that no one is asking for.

- Engineering Complexity: The expected user difficulty or effort needed to get proper results. End users of the silicon will gravitate toward the easiest, least expensive solution for them.
- Wire Efficiency: The amount of data can go down the wire, including both the critical data and the background data. Less than 100% efficiency means that the shaper injects added overhead or some amount of idle wire time to get the desired latency for the critical data.

TSN Tool	Silicon Complexity	Engineering Complexity	Wire Efficiency	Comments
Strict Priority	Low	Easy	100%	Needed component, but it is not deterministic by itself
Credit Based Shaper	Medium	Easy	100%	All CBS queues are deterministic + next highest TC (for Mgmt)
Time Aware Shaper	Medium	Hard (>1 TC) Medium (1 TC)	- Guard band - Idle opens	All TAS queues are deterministic
Frame Preemption	High	Medium but only 1 level deep	- Fragment overhead	Fragmentation can affect determinism on the other flows

Table 1: Latency TSN Tool Cost Comparison

Additional comments for Table 1:

- At the time of this writing the availability for the first three shapers in Table 1 is extremely good. The availability of the last one is not very good due to a combination of high Silicon Complexity, newness, and few customer requests. Frame Preemption is also the only standard that requires support on both sides of the wire which is another contributing factor.
- The Credit Based Shaper is the only deterministic shaper that delivers 100% wire efficiency. This is a result of its use in tandem with the Strict Priory Selector. If CBS is enabled, but there are no flows using a CBS Traffic Class, or a CBS Traffic Class is not using all its reserved bandwidth, all the unused bandwidth is available for the lower Traffic Class queues.
- The ease of using the Time Aware Shaper's is much easier when only one Traffic Class needs to be scheduled and even easier if all the critical traffic can fit in a single gate open & close per cycle. Its Wire Efficiency cannot be 100% as each time a gate window opens some guard band time is needed to insure the egress port is idle, and if a gate is open and there is no data to transmit, that bandwidth is wasted (noted as Idle Open in the table).
- Frame Preemption requires some engineering work to determine which flow, or small number of flows, can utilize it. This is because Preemption can support only 1 level, and when it occurs it influences the other flows (as they are stalled and fragmented which reduces wire efficiency). This effort is assumed to be roughly equivalent to the work needed to determine and configure the Time Aware Shaper for the same number of flows (i.e., both cases assume the use of a single Traffic Class queue with few flows). Care needs to be taken when using Preemption as its impact on the latency of the other flows is hard to determine.

TSN Shaper Performance Guidelines

The next shaper metric to consider is Performance. Regardless of what the shaper costs the user, it needs to perform the desired job in the needed time. In other words, the worst-case latency needs to be known for shapers.

As stated above, it is sometimes quite difficult to get to the absolute worst-case numbers as network topology or other parameters may come into play. And when choosing the initial shaper tool to use for a flow, a very good approximation is all that is needed. The approach taken is to modify each shaper's worst-case latency equation to be as simple as possible that results in a slightly larger latency number. In this way once the network is fully configured with all the expected streams, a network analysis tool can be used to calculate the accurate worst-case latency numbers. In the off chance there is a problem, the next section (Tool Usage Order) will propose ways to fix the problem flows.

Per hop latency equations for each shaper are shown below. Per hop is easier to evaluate as a flow may traverse 3, 5, 9 or some other number of hops. Only the network designer knows how many hops and what the maximum bounded latency target is for each flow.

<u>The Credit Based Shaper's</u> per hop first order latency equations are (as multiple Classes are supported):

- Class A ≈ $_t$Interval + $_t$MaxFrameSize
  - $_t$Interval: The observation interval of the Class (125 uSec as specified in AVB – but this can be changed for engineered networks!).
  - $_t$MaxFrameSize: The maximum size of an interfering frame + gaps, etc.
  - This simplified equation results in slightly higher number than the equation in IEEE 802.1BA-2011 subclause 6.5.
- Class B ≈ $_t$Interval + $_t$MaxFrameSize + tTimeForAllHigherFrames
  - $_t$TimeForAllHigherFrames: The time to transmit all Class A frames (+ gaps, etc.) for the duration of Class B's $_t$Interval (which is typically a multiple of Class A's $_t$Interval).
- Class C ≈ $_t$Interval + $_t$MaxFrameSize + tTimeForAllHigherFrames
  - $_t$TimeForAllHigherFrames: The time to transmit all Class A & Class B frames…
- Etc.

Link speed affects some of these parameters, but not the observation interval ($_t$Interval). Therefore, results will be calculated for both Fast Ethernet and Gigabit Ethernet. For easy comparison between shapers, the following parameters are used for all the shaper equations:

- The maximum size of an interfering frame + gaps is 1542 (1522+20) bytes.
- The size of the target frame whose latency is being calculated is 64 bytes.
- The observation interval is 125 uSec.
- And all numbers are rounded up to the next highest uSec or 10^{th} of a uSec.

<u>Example Credit Based Shaper</u> numbers are:

Class A @ FE ≈ 125 uSec + 124 uSec = 249 uSec per hop
Class A @ GE ≈ 125 uSec + 13 uSec = 138 uSec per hop

These numbers may be a surprise to you as they were a surprise to the TSN group when the latency numbers were first explored for the AVB profile (IEEE 802.1BA). Why didn't Gigabit deliver $1/10^{th}$ the latency of Fast Ethernet since it is 10 times faster? That is because the 10 times larger bandwidth of Gigabit was used to support 10 times the number of flows instead of being used to reduce the latency!

The Credit Based Shaper spreads out the reserved flows with large enough gaps such that they occupy nearly the full observation interval. With one flow this doesn't happen, but if the full 75% link capacity is fully loaded (the worst case), then on any given cycle, the target frame could be the first in the interval or the last (i.e., near the end of the observation interval). Thus the 125 uSec delay for both FE & GE. GE is faster with its lower max size interfering frame time (124 vs. 13 uSec) such that GE is nearly twice as fast per hop.

Lower latency numbers for GE are possible by using a shorter observation interval. But this number can't go below $_{t}$MaxFrameSize. And if flows cross between GE & FE links, this number can't be lowered as it is already at FE's $_{t}$MaxFrameSize and the observation interval for a Class needs to be consistent throughout the network.

<u>The Time Aware Shaper's</u> per hop first order latency equation is:

- Store & Forward with Traffic Class Gate Open ≈ $_{t}$Device + $_{t}$FrameSize
  - $_{t}$Device: The delay through a Store & Forward bridge.
  - $_{t}$FrameSize: The size of the frame passing through the bridge.

Device delay unfortunately is product specific, so you will have to ask your supplier for the accurate numbers. To advance the latency discussions in IEEE 802.1, a "rule-of-thumb" default was proposed[3]. The default used is: 2 x 64 byte times + cable delay. This works out to be 10.5 uSec for FE and 1.5 uSec for GE.

<u>Example Time Aware Shaper</u> numbers are:

Store & Forward TAS w/Gate Open @ FE ≈ 10.5 uSec + 5.2 uSec = 15.7 uSec per hop
Store & Forward TAS w/Gate Open @ GE ≈ 1.5 uSec + 0.5 uSec = 2.0 uSec per hop

[3] http://www.ieee802.org/1/files/public/docs2011/new-pannell-latency-options-0311-v1.pdf.

Gate close, transitioning to open, numbers are not interesting as the output latency numbers are zero (+/- some clock error & other uncertainties – but this should all be under 0.1 uSec). This is because the standard specifies that the Gate Open time is the time the frame's Start of Frame Delimiter (SFD) is to egresses the physical port. To do that the hardware needs to open its internal gate earlier to have that effect. But even in this case, the frame must be in memory ready to be transmitted, and that results in the same equation!

One major advantage of the Time Aware Shaper is that it can deliver even lower latencies when it is used in combination with Cut-through[4], a technique used quite often in Industrial applications. The combination of these two has been referred to as Synchronous Cut-through, but be aware, there is no official IEEE 802 standard for Cut-through, at least not yet. Cut-through is the opposite of Store & Forward in that frames can start egressing a bridge before it is fully received (i.e., Storing part). This technique greatly reduces the latency of large frames but could have no benefit for small frames. Since the target frames size being used here is the minimum of 64 bytes, we can assume Cut-through will not result in better numbers.

<u>Frame Preemption's</u> per hop first order latency equation is:

- Store & Forward with Preemption ≈ $_t$Device + $_t$FrameSize + $_t$Framelet
  - $_t$Device: The delay through a Store & Forward bridge.
  - $_t$FrameSize: The size of the frame passing through the bridge.
  - $_t$Framelet: 127 bytes + overhead if 64-byte fragmentation is enabled.

The first two parameters are identical to the ones discussed above.

The last parameter, $_t$Framlet, is the largest part of a frame that can't be preempted. If 64-byte fragmentation is being used, then 127 bytes cannot be further fragmented as it would result in a fragment smaller than the minimum 64 byte requirement[5]. This number increases when the fragmentation size is increased. Set $_t$Fragment to be (double the fragment size) – 1 if this is the case.

<u>Example Frame Preemption</u> numbers are:

Store & Forward Frame Preemption @ FE ≈ 10.5 uSec + 5.2 uSec + 11.8 uSec = 27.5 uSec per hop
Store & Forward Frame Preemption @ GE ≈ 1.5 uSec + 0.5 uSec + 1.2 uSec = 3.2 uSec per hop

---

[4] Cut-through is also covered in the presentation referenced above.

[5] Since each fragment has added overhead the minimum size frame that can be fragmented is a bit smaller than 127. 127 is used here because it is easier to see the 2 x 64-byte problem and since these are first order equations. Also see http://www.ieee802.org/1/files/public/docs2011/new-avb-pannell-latency-options-1111-v2.pdf.

Store & Forward numbers are listed as this is the general use case. Preemption used in combination with Cut-through[6] has been referred to as Asynchronous Cut-through. While this combination could be interesting for infrequent, extremely critical, extreme low latency, asynchronous communications, again there is no official IEEE 802 standard for Cut-through, at least not yet.

TSN Shaper Cost + Performance Rankings

Table 2 combines the Costs (minus the Silicon Complexity column) and the Performance data for each of the TSN Shapers discussed. A proposed Ranking for the tool usage order is also included.

TSN Tool	Engineering Complexity	Wire Efficiency	Worst Case Latency for the Examples	Ranking
Time Aware Shaper	Hard (>1 TC) Medium (1 TC)	- Guard band - Idle opens	15.7 uSec per FE Hop 2.0 uSec per GE Hop	2
Frame Preemption	Medium but only 1 level deep	- Fragment overhead	27.5 uSec per FE Hop 3.2 uSec per GE Hop	3
Credit Based Shaper	Easy	100%	249 uSec per FE Hop 138 uSec per GE Hop	1
Strict Priority	Easy	100%	Easy	Not Applicable

Table 2: Latency TSN Tool Comparison in Lowest Latency Order

It is easy to see that both the Time Aware Shaper and Frame Preemption are much faster compared to the Credit Based Shaper, but the Credit Based Shaper is Ranked first. Why? For the following reasons:

- It is fast enough for many applications.
- It is the only shaper on the list that is easy to use.
- It is the only shaper on the list that allows 100% of the wire's bandwidth to be used.
- And more than one Traffic Class can be used with different observation intervals/latencies. The other shapers are best limited to a single Traffic Class or their ease of use becomes hard.

The more than one Traffic Class bullet for CBS makes it quite is interesting. For example, standard AVB Class A audio traffic can be used in one Traffic Class queue, while at the same time as other Traffic Class queues can be configured for longer latency CAN and/or LIN traffic data types (i.e., data with equivalent message bandwidth & latency requirements) all on the same wire.

The Time Aware Shaper (TAS) is ranked above Frame Preemption for the following reasons:

- It has the absolute lowest attainable latency.

[6] https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/d2-03_pannell_increasing_network_efficiency_by_combining_ethernet_tsn_standards.pdf

- Both shapers have an impact on the other flows, but TAS's effect is deterministic while preemption's is not as easy.
- The determinism gets worse if both TAS & preemption are used at the same time. What happens if critical (scheduled) TAS data gets preempted and the frame no longer finishes before the gate window closes. There is no way to know ahead of time as preemption is asynchronous to TAS[7].
- Frame Preemption can't be used unless both sides of the wire support it.
- Frame Preemption is not widely available.

Tool Usage Order – A Proposal

Based on the Ranking of the TSN Shaper tools, the following is a proposal for a shaper selection order that can be used for each of the streams in a network. The goal is to use multiple Credit Based Shapers (CBS) with different observation intervals/latencies for all the flows & only use the other shapers if absolutely needed.

The first step is to know what is going on in the network.

- Create a sorted list of all the critical flows in the network. Place them in order from their smallest to highest allowed end-to-end latency needed for their target application.
    - Best effort flows don't need to be considered.
- Verify that the total bandwidth through any link is not loaded with more than 75% bandwidth coming from the critical flows.
    - The 75% number could go higher, but 60% to 75% is a good place to start.
    - If this number is above 75% on a link something must be done. Move flows out of the problem link to a link or links with much less load. If this is not possible, the only other option is to increase the link's speed.
- Calculate the CBS observation interval needed to meet the end-to-end latency for each flow. The number of hops and their speeds needs to be considered. Resort the list in lowest to highest observation interval order.

Examine your resorted list to find any natural groupings of similar observation intervals. Four or five sets is a great number as a lot of devices support enough Traffic Class queues for this quantity. Some interesting cut-off limit possibilities are 125 uSec, 250 uSec, 1000 uSec, 2000 uSec, etc. The 125 uSec cut-off point for the observation interval is an important one. Are there any flows that require less than this number? If so, these will need some more work as follows:

- If the flow passes only through Gigabit links, the observation interval can be reduced to 13 uSec but 31.25, or 62.5 uSec are more common lower limits.
- If the flow passes through any Fast Ethernet links, can the number of hops be reduced and/or can any of the link speeds be increased? If no, then use the Time Aware Shaper next & then Preemption as last resorts.
    - Remember, these options are limited resources that are less wire efficient & work the best supporting the fewest number of flows.

[7] Non-scheduled TAS data, by definition, is not as critical (i.e., it could be CBS data) and can be preempted.

    - Subtract any wire efficiency loss as used bandwidth against the 75% critical flow limit that is allows on a port.

After finding solutions for the hopefully very few ultra-low latency flows, process the remaining groupings in lowest needed observation interval to highest order.

- For each grouping select an observation interval that is as large as possible but that is lower than what all the flows in that grouping require.
- Start by loading each Class with no more than 20% of the links remaining bandwidth. Adjust this if needed, but remember the total for ALL critical flows & their frame overhead (IFG & preamble) along with any bandwidth & overhead used for the Time Aware Shaper and/or Preemption on the port, must not exceed 75% of any one link's bandwidth!
    - If this happens, try an alternate path.
    - Increase the observation interval where possible as this may allow more flows.

Don't forget to reserve a Traffic Class queue for Network Management (it must be the highest non-CBS Traffic Class) and one for best effort flows.

Summary and a Proposed Queueing Model

The IEEE TSN standards are designed with the intent that they can work together and that they are applicable well beyond their initial target application[8]. The Credit Based Shaper is a prime example of this as the standard supports more than the two Traffic Classes that are called for in the Audio Video Bridging profile (i.e., CBS is not limited to just audio & video data and it's not limited to the AVB profile's plug-and-play parameters).

Automotive networking has many new data delivery/latency requirements and TSN has been designed to support them all together on the same wire within the bounds of physics. Even though Automotive networks are engineered, the TSN tools enable the hardware to enforce the needed guarantees to make the engineering job much simpler.

The goal of this article was to show the interworking possibilities of the TSN Shapers and to propose a way to use them. In addition, a proposed queueing model shown in Figure 3. In IEEE 802.1 there is a current limit of only 8 Priority Code Points (PCP) that can be indicated in Tagged frames. Thus, this very limited resource needs to be used with extremely efficiency. In TSN, each PCP value becomes more like a Class-of-Service label instead of a priority value. Their assignment in the figure reflects this approach and it is consistent with the AVB standards so the audio & video data types can be supported too.

Four Credit Based Shapers are shown so that multiple widely variable observation intervals can be used as the procedure above described. The lower Traffic Classes are reserved for Scheduled (TAS) and Express (preemption) traffic if these are needed. And the also important Network Management (shown as the highest non-CBS Traffic Class) and best effort queues are included.

---

[8] While this approach is great for the future applicability of the standards, it sometimes makes the standards hard to read.

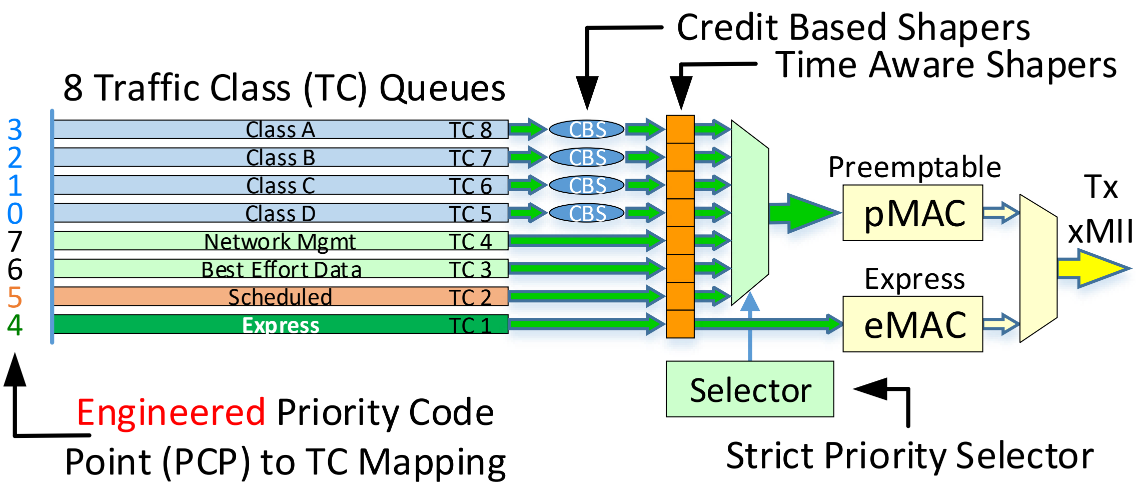

Figure 3: Proposed TSN Egress Queue Model

This is a hard concept that has been simplified so that an easy starting point on which shaper to use for a target flow can be made. Thus, the listed latency numbers are in the correct range, but they are still high-side approximations. Therefore, tools need to make final calculations using available equations from the standards and from experience.

Standards writers are only human and on some rare occasions the intent is not reflected in the standard. This is where you can help. If you find problems with an IEEE 802.1 standard, or just what to learn more, you are always welcome to attend any IEEE 802 meeting or conference call as they are all public. There is also a Maintenance process to report issues.