Introduction to Time-Sensitive Networking

Norman Finn

Abstract

A number of companies and standards development organizations have, since 2000, been producing products and standards for "time-sensitive networks" to support real-time applications that require a) zero packet loss due to buffer congestion, b) extremely low packet loss due to equipment failure, and c) guaranteed upper bounds on end-to-end latency. Often, a robust capability for time synchronization to less than 1 μs is also required. These networks consist of specially-featured bridges that are interconnected using standard Ethernet links with standard MAC/PHY layers. Since 2012, this technology has advanced to the use of routers, as well as bridges, and features of interest to time-sensitive networks have been added to both Ethernet and wireless standards.

Introduction: Three Kinds of Packet Service

Best effort packet service is familiar to users of routers and bridges. It delivers most packets, most of the time, mostly in order. There are no guarantees. Performance is statistical. If one plots (Fig. 1) the probability density of delivery, end-to-end latency, or variation in latency over a given time interval, one sees long, low-probability tails on every curve.[1]

Constant bit rate (CBR) service is typically offered by time-division multiplexing (TDM) facilities such as synchronous digital hierarchy (SDH) or optical transport network (OTN). Latency is fixed, and jitter is essentially zero (Fig. 2). The service offers connections; every packet flows end-to-end through the connection. The packet loss curve shows that CBR eliminates congestion loss, so is almost zero if the proper buffering is present. If we assume that 1:1 or 1+1 protection is used,[2] packets are lost at a low rate, but in large groups, when an equipment failure is detected and an alternate path activated.

Time-sensitive network (TSN) service, the subject of this article, is another kind of service that is gaining users and market attention. It is based on a best-effort packet network, but the network and an application have a contract. This contract limits the transmitter of a TSN flow to a certain bandwidth (maximum packet size and maximum packets per time interval). The network, in return, reserves bandwidth, buffering, and scheduling resources for the exclusive use of these TSN traffic flows. For these flows, the contracts offer bounded latency and zero congestion loss. In addition, packets belonging to a flow can be sequenced and delivered simultaneously along multiple paths, with the duplicates deleted at or near their destinations. The curves for this service are shown in Fig. 3.

The biggest difference between CBR (Fig. 2) and TSN (Fig. 3) is that the latency and latency variation curves have a larger range, though they are still bounded. The packet loss curve for TSN service has a much lower tail than the CBR curve, because TSN uses a different protection scheme (discussed later in the section "Packet Replication and Elimination") than the 1:1 protection usually employed in CBR.

Some applications are a natural fit with constant bit rate (CBR) service. The original CBR services, telephony and telegraphy, are obvious examples. Some applications are a natural fit with best-effort packet service. Web browsing is typical of this usage.

Some applications, however, have never been able to use best-effort service. Examples are industrial control, audio and video production, and automobile control. When these industries moved from mechanical or analog technologies to digital technologies in the 1980s, best-effort packet technologies, including Ethernet, were not suitable, so these industries had to invent special-purpose digital systems. The problem with Ethernet included its high cost, compared to special-purpose digital connections, and its inherent unpredictability. Collision detection and retransmission algorithms were not suitable for physical control systems.

Networking technology is now at the point where best-effort networking equipment can, at a modest cost, supply TSN services, in addition to its best-effort services, that meet the needs of many applications that formerly required either CBR service or special-purpose digital connections. Because of the huge increase in the demand for networking, Ethernet is now cheaper than special-purpose digital connections, so there is significant incentive for these industrial and control applications to migrate to Ethernet.

Table 1 gives an overview of the essential differences between CBR, best-effort with typical QoS features, and TSN services. These are not three specific classes of service, but ranges of features supplied by each type of packet service.

Essential Features of TSN Networks

TSN is a feature supplied by a TSN network that is primarily a best-effort packet network consisting of bridges (and, in the future, routers or network appliances). The TSN quality of service is supplied

[1] End-to-end latency and latency variation are per packet. Loss probability is highest if few buffers are allocated, but still finite with many buffers allocated.

[2] See IETF RFC6718 for a description of 1:1 and 1+1 path redundancy in a similar context.

Digital Object Identifier: 10.1109/MCOMSTD.2018.1700076

The author is with the California Institute of Technology.

Networking technology is now at the point where best-effort networking equipment can, at a modest cost, supply TSN services, in addition to its best-effort services, that meet the needs of many applications that formerly required either CBR service or special-purpose digital connections.

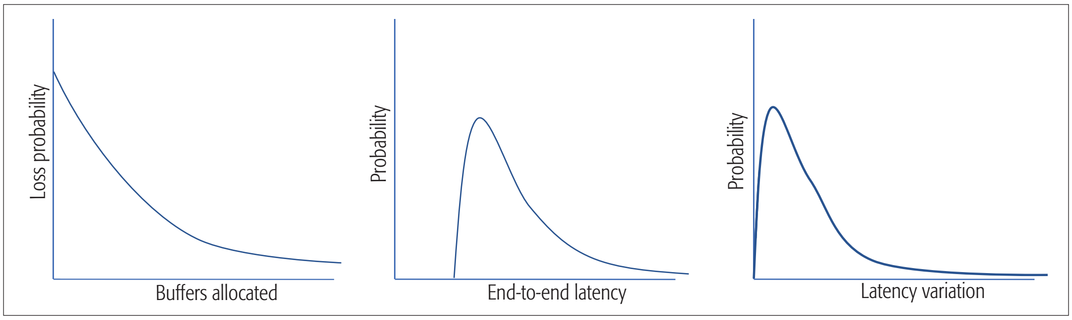

FIGURE 1. Best-effort packet service.

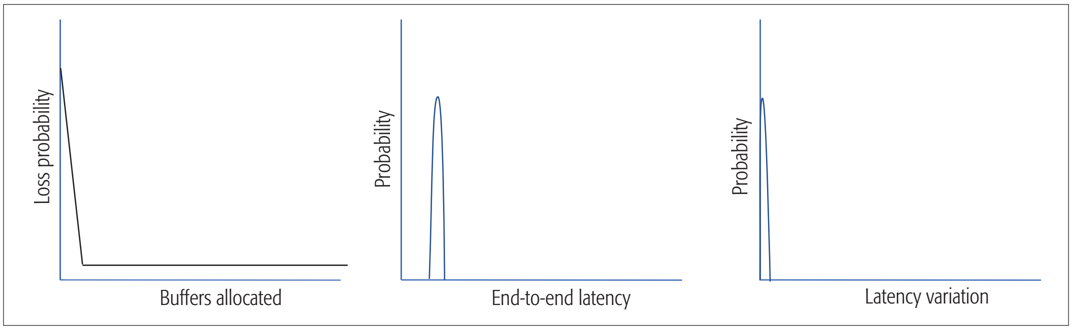

FIGURE 2. Constant bit rate packet service.

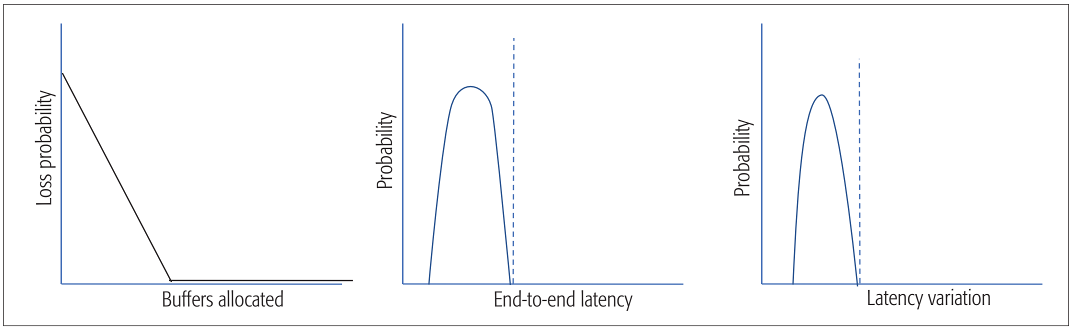

FIGURE 3. TSN packet service.

to "TSN flows" designated as being critical to a real-time application.

The essential features of TSN networks are:

1. Time synchronization. All network devices and hosts can synchronize their internal clocks to an accuracy between 1 μs and 10 ns. Synchronization is accomplished using some variant of the IEEE 1588 Precision Time Protocol.

2. Contracts between transmitters and the network (discussed later in the section "Control and Managemen"t): Every TSN flow is the subject of a contract arranged between the transmitter of the flow and the network. This enables TSN networks to provide:

a) Bounded latency and zero congestion loss. Congestion loss, the statistical overflowing of an output buffer in a network node, is the principle cause of packet loss in a best-effort network. By pacing the delivery of packets and allocating sufficient buffer space for TSN flows, congestion is eliminated, and any given TSN flow can be promised a worst-case latency for delivering its packets end-to-end through the network.

b) Ultra-reliable packet delivery. Having eliminated congestion loss, the next most important cause of packet loss is equipment failure. TSN networks can send multiple copies of sequence-numbered data flows over multiple paths, and eliminate the duplicates at or near the destinations. There is no cycle of failure detection and recovery, as every packet is duplicated and taken to or near its destinations, so that one equipment failure does not cause the loss of even one packet.

c) Flexibility. New contracts can be made and old ones ended. As TSN flows come and go, the proper functioning of all TSN flows is maintained at all times.

3. Coexistence with best-effort services. Unless the demands of the TSN flows consume too much[3] of a particular resource, such as the bandwidth of a particular link, TSN traffic can be paced so that the customary best-effort Quality of Service practices such as priority scheduling, weighted fair queuing, random early discard, etc., still function in their usual manner, except that the bandwidth available to these capabilities is reduced by the TSN traffic. Furthermore, data not subject to a TSN contract ("non-TSN" traffic) can use any contracted bandwidth unused by a TSN flow.

[3] "Too much" has no fixed definition. IEEE 802.1 has used 75% as a design goal for the upper limit to the proportion of traffic that is Deterministic.

Characteristic	Constant bit rate	Best-effort	TSN
Connectionless?	Connections only	Connectionless	Allocated resources along fixed paths
End-to-end latency	Constant for all flows sharing a path	Statistical: subject to semi-random fluctuations due to congestion or equipment failure	Bounded: latency cannot exceed a specified maximum
In-order delivery	In-order	In-order delivery except when network topology changes	In-order delivery except when lost packets are recovered
Latency variation	Essentially zero	Statistical, often low	Limited by minimum latency and bounded latency
Response to equipment failure	Detect failure, switch to alternate path (1:1 or 1+1)	Detect failure, propagate new topology, alter local routing decisions	Packet replication and elimination: no failure detection or response mechanism
Primary causes of packet loss	Random events (cosmic rays, signal/noise) or equipment failure	Congestion: momentary overflow of output queue	Equipment failures exceeding the number of redundant paths
Granularity of packet loss	Packets are lost in groups whenever equipment fails or is repaired	Random, relatively high probability	Packets are lost only as long as excessive equipment failures persist
Penalty for sending excess data	Excess data lost; no effect on other flows	Depending on QoS used, excess may or may not affect other flows	Excess data lost; no effect on other flows
Unused contracted bandwidth	Lost	Available to all flows, with or without contracts	Available only to non-contract flows

Table 1. Three types of packet service.

The reader should note that item 2c above, flexibility, is the most radical change to most existing paradigms for supporting real-time applications over best-effort networks. Other alternatives to TSN (discussed later in the section "Alternatives to TSN") require network simulation, prototyping, and/or run-time testing to determine whether a change to the critical flows can or cannot be supported. Changes can only be made to such real-time networks when the applications are down. TSN networks can be built to support a dynamic environment.

Use Cases for TSN

Use cases targeted by the TSN standards include:

- Professional audio and video studios.
- Electrical power generation and distribution.
- Building automation.
- Cellular radio: interconnecting the data baseband processing and radio frequency blocks of a cellular radio base station (fronthaul).
- Industrial machine-to-machine: closed-cycle control loops, employing measurement, computation, and command sub-cycles.
- Automotive and other vehicle applications.
- Service provider: CBR service over best-effort networking equipment.

Time Synchronization

The natural paradigm for dedicated digital busses is, "Do what the packet says to do when you receive the packet." Timing is synchronized by the clock in the controlling device in the network, and the reception of the data it transmits. Transmission times are short and perfectly predictable.

Given that TSN uses a network, and that the cost of that network depends upon the degree to which the timing is fixed, the natural paradigm for TSN is, "Do what the packet says to do at the time the packet says to do it." Time is then synchronized separately from the data flow; the only requirement is for an upper bound on end-to-end latency, so that the packet can be delivered before its intended execution time has passed. Thus, time synchronization is required for many applications, so it is considered a part of TSN. However, synchronization is separable from the rest of TSN, in that none of the TSN features are tied to any particular means for synchronizing time. The Precision Time Protocol, whose root definition is in IEEE Std 1588, is the typical means for synchronizing the network's clocks, but other algorithms can be used, if they meet the accuracy requirements of the user's application.

Zero Congestion Loss, Bounded Latency

Never an Empty Buffer

Let us assume for a moment the usual case for Ethernet, that a physical link transmits packets serially, rather than transmitting more than one packet simultaneously, that a packet contains the address of its destination(s), which is used by a forwarding node to select an output port, and that a packet is received and checked for errors by a forwarding node before it is transmitted to the next node or to the destination. Then, every packet must be stored in a buffer for some period of time at each forwarding node.

In the timing model employed for TSN, each output port has some number of first-in-first-out (FIFO) queues associated with it for temporary storage of packets that are to be transmitted on that port. Each incoming packet is examined, and assigned to one (or more) output ports for transmission. On each port, it is also assigned to a particular queue. As each packet finishes transmission, one or more queue selection algorithms (discussed later in the section "Queuing Algorithms") cooperate to select a packet from one of the port's queues to transmit next.

There are two reasons why the rate at which packets enter a given queue can vary. First, any variations in the processing of the packet for forwarding will cause variations in the arrival rate to the queues. Delay variation requires storage. If the delay time through a path is growing, the amount of data stored in the path is growing, so the arrival rate at the next hop is slower, and if the delay time is shrinking, the stored data is being dumped, and the arrival rate is higher. Second, the queuing algorithms used in the immediately preceding hops may cause bursts or gaps in the rate at which any given flow's packets are transmitted. There is, by definition, a rate at which any given flow is transmitted, but the timescale over which that rate is measured can be many packet transmission times in length, so small-scale variations are common.

When the input rate to a queue exceeds the output rate for a sufficient length of time, the queue must overflow. This is congestion loss, and this is what TSN seeks to avoid.

Imagine a complete saturated TSN network, in which all data is critical, and 100% of each link's bandwidth is allocated to some number of TSN flows using that link. Every source is transmitting at exactly its allotted rate. The flows traverse the network in all directions; no two flows take exactly the same path through the network. Imagine that there are no variations in the forwarding delay.

Imagine, now that one flow, flow A, stops. On some output port through which flow A was passing, when the transmission opportunity for one of flow A's packets comes up, the node must either output nothing, or output a packet belonging to some other flow B. If it outputs a packet from flow B, then in the long term, it is exceeding the normal rate for flow B, and runs the risk of overflowing the queues for flow B in the next hop. With sufficient analysis, it may be possible to determine the limits for how much excess data in flow B, or flow C, from this and from other ports feeding the next hop, can be accommodated before causing an overflow.

However, this analysis is very difficult. TSN avoids the analysis by transmitting nothing (or transmitting a non-TSN packet) when it has nothing to transmit for a given flow. This leads to TSN making the following requirement for network nodes:

For every flow traversing a forwarding node, sufficient data is buffered in a forwarding node to ensure that a transmission opportunity for that flow is never missed, unless the source of the flow slows or stops. That is, for every flow, at every hop, over some finite time scale, the input rate equals the output rate.

Buffer Allocation

The only way to provide zero congestion loss is to be able to predict the worst-case buffer requirement. This is possible because:

- Input rate = output rate for each flow (see the previous section).
- Each queuing algorithm suitable for TSN (see the next section) defines its own mathematically analyzable packet selection schedules.

The worst-case variations from one hop's output can be set against the worst-case variations in the next hop's output to get the number of buffers in the next hop required for queue selection variation. These can be added to the worst-case forwarding delays to get the number of buffers that must be allocated to the queue.

Queuing Algorithms

The queuing algorithms discussed below are defined by IEEE 802.1, mostly in IEEE Std 802.1Q, Bridges and Bridged Networks.

Credit Based Shaper (CBS): Defined by clause 35 of IEEE Std 802.1Q-2014. This shaper outputs packets at a rate such that, over a relatively short term, is equal to the total bandwidth allocated to the TSN flows using that queue. The worst-case delay that the CBS queues can impose on the highest-priority best-effort queue (which is always lower priority than all of the CBS queues) is computable.

Time-Scheduled Queues: Defined by IEEE Std 802.1Qbv. All queues (not just the TSN queues) on a given port are attached to a rotating schedule, which in turn is regulated by a clock synchronized with the other bridges. The network manager can set queue-on and queue-off events on a 1-ns granularity, although an implementation may be less accurate in practice.

Creating a schedule for a set of applications is not a trivial computational task. For example, a network schedule may be constrained to provide frequent transmission opportunities for non-TSN traffic, so that the most important best-effort flows (VoIP or routing protocols) can achieve their latency goals. Time-scheduled queues are very flexible; they can be used to achieve goals very different from TSN bounded latency.

Transmission Preemption: Defined by IEEE Std 802.1Qbu and IEEE Std 802.3br. These standards allow some queues on an output port to be designated by network management as "preemptable" and others as "preempting." Packets that have started transmission from a preemptable queue can be interrupted if a preempting queue is selected for transmission. Transmission of the preempted packet is resumed from the interruption when there are no more preempting queues selected for transmit. This reduces the degree to which non-TSN packets can interfere with the transmission of TSN packets.

Input Scheduling and Cyclic Queuing and Forwarding (CQF): Defined by two standards, IEEE Std 802.1Qci and IEEE Std 802.1Qch. A rotating schedule, as for time-scheduled queues, above, is defined both on input ports and output ports. The net result is that packets progress through the network in groups, stopping for exactly one output schedule cycle time at each hop.

Asynchronous Traffic Shaping (ATS): This is a new project, IEEE P802.1Qcr. The intention is to have a mechanism that gives better overall latency than CQF, but is not too much more expensive to implement.

Packet Replication and Elimination

IEEE Std 802.1CB contains a very complete introduction to Frame Replication and Elimination for Reliability (FRER) in its clause 7. The essential features of FRER (see Figure 4) are:

- To every packet from a source, or in a particular flow, a sequence number is added.
- The packets are replicated, creating two (or more) identical packet flows. These flows can be unicast flows or multicast flows.

For every flow traversing a forwarding node, sufficient data is buffered in a forwarding node to ensure that a transmission opportunity for that flow is never missed, unless the source of the flow slows or stops. That is, for every flow, at every hop, over some finite time scale, the input rate equals the output rate.

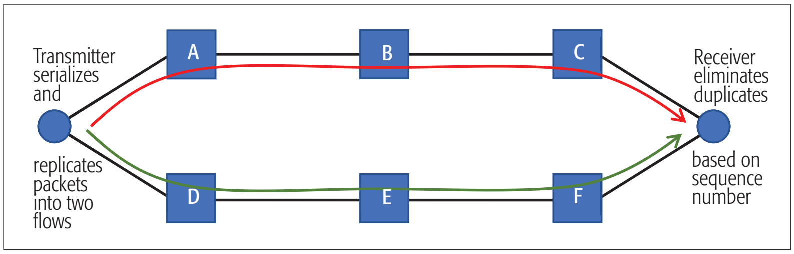

FIGURE 4. Packet replication and elimination.

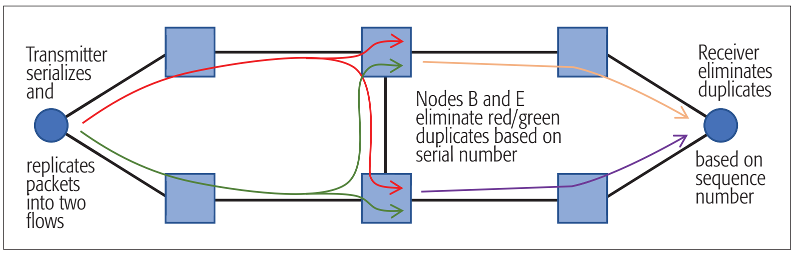

FIGURE 5. Packet replication and elimination.

- At some point at or near the receiving end station(s), the duplicate packets are detected and eliminated.

This technique is similar to the typical 1+1 scheme used by CBR technologies, differing in that a 1+1 receiver selects one stream to pass and one to ignore, while FRER makes a packet-by-packet choice. FRER is proof against any singe failure in the network. Of course, the transmitting and receiving stations themselves are single points of failure, but many applications can provide redundant transmitters and receivers to overcome this.

- The network can also be configured to discard and re-replicate packets at various points in order to be able to handle multiple failures.

We can see in Fig. 5 that some two-failure events, such as the failure of A and F, B and C, or D and F, will not cause the loss of a packet. A failure of both A and D, however, would stop the flow. More complex configurations are possible.

In-order delivery is not required by the standard because bulk flows could require network nodes to have large buffers to put the packets back in sequence.

This is also exactly the technique described by ISO/IEC 62439-3 (discussed later in the section "Other Relevant Standards). In fact, 62439-3 predates the work on IEEE 802.1CB. Both standards operate only at Layer 2 with bridges. Both use 16-bit sequence numbers.

Control and Management

The IEEE TSN standards provide a rich set of management controls for creating a TSN network, starting with the plug-and-play profile IEEE Std 802.1BA, which allows some applications (audio/video, in particular) to be supported without requiring a knowledgeable network administrator. In such a network, the senders, receivers, and bridges use the protocols defined in IEEE Std 802.1Q to make bandwidth reservations. FRER is not supported.

There are four aspects to the use of control and management to fully utilize TSN capabilities in a network:

•A selection of bridges and end stations must be made, including a selection of capabilities for each device. The physical topology of the network must be established.

•The static characteristics of the network must be defined. Examples of the choices to be made include the selection of topology control protocols used for non-TSN traffic, and perhaps for TSN traffic as well, the maximum bandwidth to be allocated for TSN flows on each physical link, and a selection of protocol(s) and parameters for time synchronization. One may statically create subsets of the physical topology to be used for FRER paths.

•One or more means for creating bandwidth reservations must be selected, and then employed. Reservations can be controlled statically by management action. They can be controlled through the action of application end stations, bridges, or management stations, using the reservation protocols defined in IEEE Std 802.1Q and/or IEEE Std 802.1Qca.

•An architecture for the dynamic control of flow reservations must be selected, if required by the application(s) using TSN. Each application end station and bridge can make independent reservation decisions using the selected protocols. The bridges can defer reservation decisions to a central network controller. Often, there exists an "application controller" entity that controls a time-sensitive application. This controller can be a part of the network controller, it can order the end stations to make reservations, or it can ask the network controller for reservations on behalf of the end stations. (See IEEE P802.1Qcc.)

The standards to fully support some of the above options are still under development in IEEE 802.1.

Alternatives to TSN

Some real-time systems have been controlled using Ethernet since the technology was invented in the 1980s. The alternatives often employed include:

- Overprovisioning: One builds a network that has significantly more physical link bandwidth and buffer space than is required by the critical data.
- Isolation: A network, physically isolated from other networks, is constructed for the exclusive use of one or a small number of critical applications.
- Prioritization: Critical data is given the highest priority in the network, perhaps even higher than network control traffic, to minimize the impact of non-critical on critical traffic.
- Weighted fair queuing (WFQ) and similar prioritization schemes: Bandwidth and resources can be allocated to be statistically fair among critical flows.
- Congestion detection: This technique typically causes a flow that is experiencing congestion to slow down. This is not applicable to the applications for which TSN is designed; these applications cannot slow down the real-time physical world to accommodate the network's fluctuating load.
- Congestion avoidance: Routing flows over less-congested network paths. This can work at the time a flow is established, but afterwards, its efficacy declines.

The techniques listed above suffer from one or more of the following difficulties, compared to TSN:

1. Statistical vs. deterministic: Most of the above techniques reduce the probability of packet loss or late delivery, but most do not prevent it absolutely. An engineer must balance the degree of overprovisioning against the probability of late delivery or packet loss.

2. Predictability: In all of the schemes above, the only way to determine whether a given mix of critical flows will achieve the required level of reliability is to try the application and see if it works, either by simulation or by actual experiment.

3. Corner cases: Only the most detailed and exhaustive simulation exercises can give one confidence that there is no corner case, when just the wrong processes are turned on or off at just the right moment, that will disrupt some critical flow.

4. Dynamism: The lack of predictability means that, for the most part, changes to any application or to the network can only be performed when the network is not in use. Every dynamic choice accommodated (e.g. turning an application on or off) increases the simulation and testing load exponentially.

5. Standardization: There exist a number of Ethernet-based solutions employing proprietary techniques for the network nodes and/or the MAC/PHY hardware, that solve many or all of the problems addressed by TSN. These proprietary solutions are generally more expensive for the customers than solutions based on open standards. (Some of these techniques are, in fact, being included in the TSN standards.)

6. Expense: Strict isolation, i.e., one network per application, in combination with the other techniques shown above, can solve all of these problems. But this solution is expensive.

Note, however, that there do exist standard techniques that can provide the low or zero congestion loss features of TSN (e.g., IETF RFC2998).

History

IEEE 802.1 created an Audio Video Bridging (AVB) Task Group in 2007. Its goal was to replace HDMI, speaker, and RCA cables in the home with Ethernet. A secondary goal was the small audio or video production studio. The standards produced supported time synchronization through an 802.1-specified profile of IEEE 1588, a reservation protocol for transmitters, receivers, and bridges to create contracts, a queue draining technique (the credit-based shaper) to enforce the contracts, and an overall profile specification that specified how to configure standard components to implement a plug-and-play home or small studio. AVB works only in bridged Layer 2 networks.

HSR/PRP: ISO/IEC 62439-3 defines the High-availability Seamless Redundancy (HSR) and the Parallel Redundancy Protocol (PRP) as well as a profile of IEEE 1588. They provide protection against packet loss due to chance or equipment failure, but do not offer bounded latency or zero congestion loss.

Demand from the industrial control community and from the automotive community led to the renaming of the IEEE 802.1 AVB Task Group to the Time-Sensitive Network (TSN) Task Group in 2012, and a broadening of its goals.

The 802.1 TSN TG has produced a number of standards. IEEE 802.1 standards are, for the most part, confined to Layer 2. That is, only bridged networks are supported, and data flows that require a router are not supported end-to-end. The TSN standards have augmented the techniques of AVB to include better reservation (contract creation) protocols, more queue draining techniques, and HSR/PRP-like packet replication. A number of standards are currently in progress, including a profile of TSN standards to enable the use of time-sensitive networking for cellular fronthaul.

In 2015, a Deterministic Networking (TSN) Working Group was created in the Internet Engineering Task Force (IETF). This group is expanding TSN concepts to include routers, so that the techniques developed in TSN can be extended to routed data flows. It also has a goal to scale up the TSN techniques so that they work in larger networks than can be supported by Ethernet bridges.

In 2015, a Deterministic Networking (TSN) Working Group was created in the Internet Engineering Task Force (IETF). This group is expanding TSN concepts to include routers, so that the techniques developed in TSN can be extended to routed data flows. It also has a goal to scale up the TSN techniques so that they work in larger networks than can be supported by Ethernet bridges.

Standards Summary

IEEE 802.1 AVB, 802.1 TSN, and 802.3 Standards

Standards listed as "IEEE Std 802.xyz-2xxx" are complete, published standards. Those listed as "IEEE P802.xyz" (note the "P") are works in progress. A given standard or work in progress can be either a stand-alone document, or an amendment to a previous standard, as indicated in the text. See the 802.1 web site (http://www.ieee802.org/1) for the most up-to-date information.

Important Note: IEEE 802 standards must be purchased from the IEEE (http://standards.ieee.org/findstds/) for the first six months after publication, and are available free from the GetIEEE web site (http://standards.ieee.org/about/get/) after that time. IEEE 802.1 works in progress are available from the IEEE 802.1 web, using a username and password, to anyone, IEEE member or not, interested in making helpful comments to further the work of the committee.

A. IEEE Std 802.1AS-2011–Timing and Synchronization
B. IEEE Std 802.1Q-2014–Bridges and Bridged Networks
C. IEEE Std 802.1BA-2009–Audio Video Bridging (AVB) Systems
D. IEEE Std 802.1CB-2017–Frame Replication and Elimination for Reliability
E. IEEE Std 802.1Qbu-2016–Frame Preemption (amendment to 802.1Q)
F. IEEE Std 802.3br–Interspersing Express Traffic
G. P802.1Qcc–Stream Reservation Protocol (SRP) Enhancements and Performance Improvements (amendment to 802.1Q)
H. IEEE Std 802.1Qbv-2015–Enhancements for Scheduled Traffic (amendment to 802.1Q)
I. IEEE Std 802.1Qca-2015–Path Control and Reservation (amendment to 802.1Q)
J. IEEE Std 802.1Qch-2017–Cyclic Queuing and Forwarding (amendment to 802.1Q)
K. IEEE Std 802.1Qci-2017–Per-Stream Filtering and Policing (amendment to 802.1Q)
L. IEEE P802.1CM–Time-Sensitive Networking for Fronthaul
M. IEEE P802.1Qcr–Asynchronous Traffic Shaping (amendment to 802.1Q)

IETF DetNet Drafts

As yet, there are no RFCs or Standards from the IETF Deterministic Networking (DetNet) working group. Internet drafts are works in progress, and quickly become out-of-date. See the DetNet documents list (https://datatracker.ietf.org/wg/detnet/documents/) for the most up-to-date list of DetNet drafts. The drafts listed here have been adopted by the DetNet Working Group.

A. Deterministic Networking Problem Statement
B. Deterministic Networking Use Cases
C. Deterministic Networking Architecture
D. DetNet Data Plane solution
E. Deterministic Networking (DetNet) Security Considerations

Other Relevant Standards

A. IEEE Std 1588-2008–Precision Clock Synchronization Protocol for Networked Measurement and Control Systems
B. ISO/IEC 62439-3:2016–Industrial Communication Networks–High Availability Automation Networks
C. IETF RFC2998–A Framework for Integrated Services Operation over Diffserv Networks

Biography

Norman Finn (nfinn@alumni.caltech.edu) received his B.S. in astronomy from the California Institute of Technology. His standards work includes ATM LAN Emulation, ITU-T Ethernet OAM, and numerous IEEE 802 projects: VLANs, Link Aggregation, Link Layer Discovery Protocol. He has edited nine IEEE 802.1 and 802.11 standards. He currently represents Huawei for Time-Sensitive Networking in IEEE 802 and in the IETF. He has been awarded more than 100 patents.