Design Aspects of Low-Latency Services with Time-Sensitive Networking

Csaba Simon, Markosz Maliosz, and Miklós Máté

Abstract

The aim of Time-Sensitive Networking (TSN) is to provide determinism to IEEE 802 networks. With TSN, data streams of general background applications can share the same standards-based network infrastructure without impacting time-critical applications, also leading to cost savings by using a single network infrastructure. Deterministic ultra-low latency increases efficiency for operations that require consistent message exchange. The most stringent delay requirements are specified for industrial and fronthaul network applications; thus, we demonstrate network engineering aspects in these two networking scenarios. In industrial networking, deterministic delays are important, especially for real-time control processes. In fronthaul networks, the timely delivery of data units with minimum delay variations is required for efficient operation. In this article, we present two mature TSN tools: time gated queuing and frame preemption. Then we highlight their operational and design aspects when using them in industrial and fronthaul networks, investigating their applicability through different simulation scenarios.

Introduction

The primary need for deterministic networking is to provide end-to-end delivery of messages within a given time. Time-critical traffic requires specific timing and availability. More and more emerging use cases require the transmission of mission-critical and noncritical traffic over a shared network. However, the de facto layer 2 standard of modern communication networks, Ethernet, was originally designed for best effort operation, which is not suitable for such goals. Over the years IEEE recognized these needs and started to enhance the IEEE 802.1 bridge features. These enhancements fall under the IEEE 802.1 Time-Sensitive Networking (TSN) Task Group [1].

Several use cases may benefit from the deterministic delays achievable with TSN enhancements. Human reaction times are on the order of 100 ms, and applications that involve or replace human actions are typically one order of magnitude faster, including augmented reality (AR), remote motion control, autonomous driving, and more [2].

There is also a growing interest in use case scenarios with ultra-low latency requirements, where end-to-end latencies must be lower than 1 ms. A number of trending applications emerge from several machine type communications, such as motion control, closed loop automation control, and other factory automation tasks in smart factories [3]. A second use case of great importance is the fronthaul networking in the communications industry. Fronthaul networks connect more radio equipment (RE) to a radio equipment controller (REC), and have 100 μs upper bound on the end-to-end latency. While new use cases may appear in the near future, the importance of these two has already been recognized; thus, in our article we focus on them.

Although the 802.1 TSN features are applicable to any networking technology within the 802 family, the use cases we consider only target IEEE 802.3 Ethernet. The presented ultra-low-latency scenarios used to be deployed over dedicated networks to support their demanding quality of service (QoS) requirements, but the application of TSN features over switched Ethernet networks offers an alternative transport infrastructure.

Migrating from specialized networking solutions to standard IEEE 802 technology may offer a versatile and cost-effective solution to the networking needs, and enable resource sharing, since classical services already using Ethernet as transport can be carried over the same infrastructure. Nevertheless, this comes with the need for proper planning. Design efforts should not only cover resource scheduling, but before that, as a first phase of the work, it is required to establish the prerequisites, such as to understand the effect of interference from other traffic, to set the traffic parameters, and to select the proper TSN features or their combination (when applicable).

During our research on various TSN tools over the last few years, we have concluded that giving sufficient attention to the first phase of the design effort has a strong impact on several areas: it might offer cost savings, and simplify the scheduling task as well as the operation of the network.

Our findings shed light on several aspects of the TSN features when ultra-low-latency requirements are set. We present simulation-based numerical results, relying on a representative network topology for each use case. Our results may aid engineers facing such latency requirements, offering the proper background to decide on the trade-offs between performance and resource usage. In our use cases there are common issues to discuss, because both require ultra-low latencies. On top of that, each scenario has its own particularities, as detailed later.

In the next section we introduce the TSN fea-

Digital Object Identifier:
10.1109/MCOMSTD.2018.1700081

1The authors are with Budapest University of Technology and Economics.

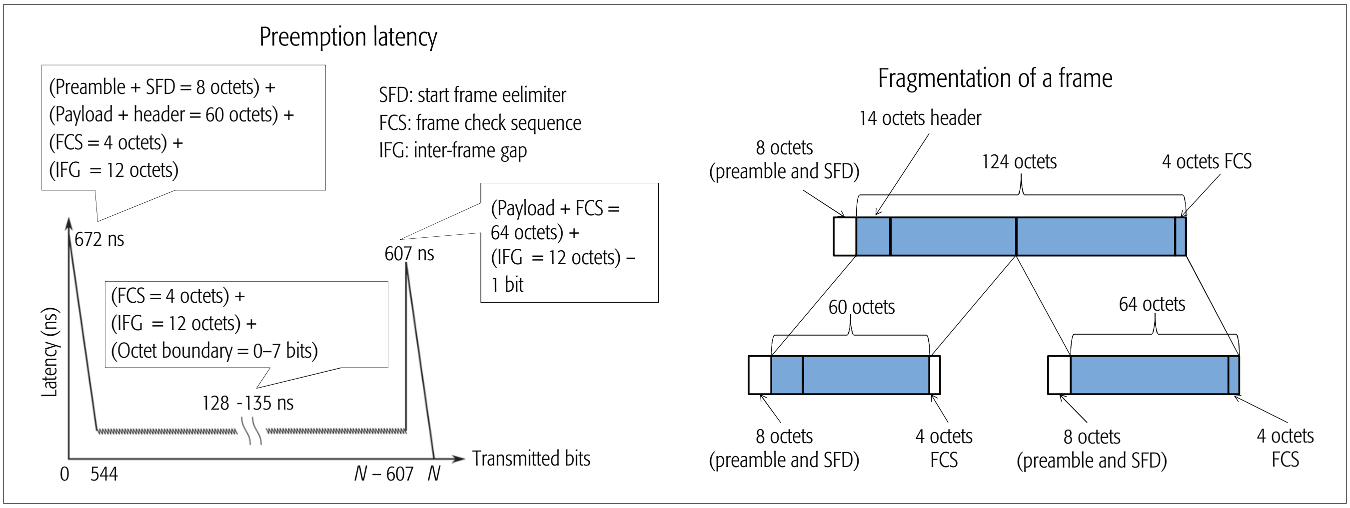

Figure 1. Left: preemption latency depending on the number of bits already transmitted of an $N > 123 \times 8$-bit-long preemptable frame on 1 Gb/s links (1 bit is 1 ns on the wire; right: the fragmentation of a 124-octet-long preemptable frame.

tures/tools used in our work. Then we present the investigated scenarios and the experience gained, and discuss the lessons learned. Finally, we conclude our article.

Time-Sensitive Networking Tools

The TSN Task Group developed several mechanisms for increased reliability and deterministic packet delays in IEEE 802.1 bridged networks. Apart from the public standards, there is a great amount of ongoing work targeting all aspects of 802.1 bridge operation [1]. In this article we focus on those public TSN standards that affect the performance of time-critical transport, namely the following two tools that extend the data plane functionality of bridges: enhancements for scheduled traffic and frame preemption.

Enhancements for Scheduled Traffic

The IEEE 802.1Qbv amendment, "Enhancements for Scheduled Traffic" [4], adds new capabilities to the eight priority queues, which were originally defined in IEEE 802.1p and merged into 802.1Q [5]. A *transmission gate* is associated with each queue; these gates are able to block the traffic of each priority class individually. The state of the gates is either open or closed, which is governed by a gate *control list* associated with the outgoing port. The gate control list is a series of 8-tuples specifying the state of the 8 gates. The list is advanced periodically by the cycle timer state machine with a configurable advance speed. We will refer to this mechanism as *time gating*, since the transmission gates are controlled by a timer according to the control list.

A frame in the outgoing queue of a port is not eligible for transmission if the gate of its priority class is closed, or if the gate is open but will close before the transmission of the frame would finish. Therefore, a conforming bridge implementation must be aware of the properties of the physical medium of its ports to be able to plan its gated transmissions.

Transmission gates were aimed at separating traffic belonging to different priority classes in time, thus eliminating interference among them. For example, when two priority classes are used, and the gates open in a mutually exclusive way, the packets of one class can never block packets of the other class from using the output link.

The resolution of the gate control list is of nanosecond precision; thus, it requires precise time synchronization across the network. To support this level of synchronization in IEEE 802.1 bridged networks, TSN includes an improved version of IEEE 802.1AS [6], which is the adaptation of the IEEE 1588 Protocol.

Frame Preemption

The IEEE 802.1Qbu amendment, "Frame Preemption" [7], and its companion IEEE 802.3br, "Specification and Management Parameters for Interspersing Express Traffic" [8], add the capability of interrupting a frame transmission to transmit a frame of higher priority. Without having to wait for the lower-priority transmission to fully finish, the express frame suffers shorter latency. The eight priority levels are split into two groups: *express* and *preemptable*. The queues assigned to priority levels belonging to the express group are referred to as express queues.

The transmission of the preempted frame resumes after the express traffic is finished, and the receiver is able to reassemble the preempted frame from the fragments. This is a radical change in the IEEE 802 architecture, where the transmission of a frame was always considered an atomic operation.

Figure 1 illustrates the preemption latency on a 1 Gb/s link when the preemptable frame is larger than 123 octets, depending on the number of bits of the preemptable frame already transmitted. In the middle of the preemptable frame, the preemption latency is 128–135 ns, depending on the waiting time for the octet boundary. If preemption happens near the end of the frame, where fewer than 64 octets remain (including the FCS), preemption of that frame is no longer possible (maximum 607 ns latency at the start of this phase). If the preemption signal arrives at the beginning of the frame transmission, the fragmentation process must wait until 64 octets are trans-

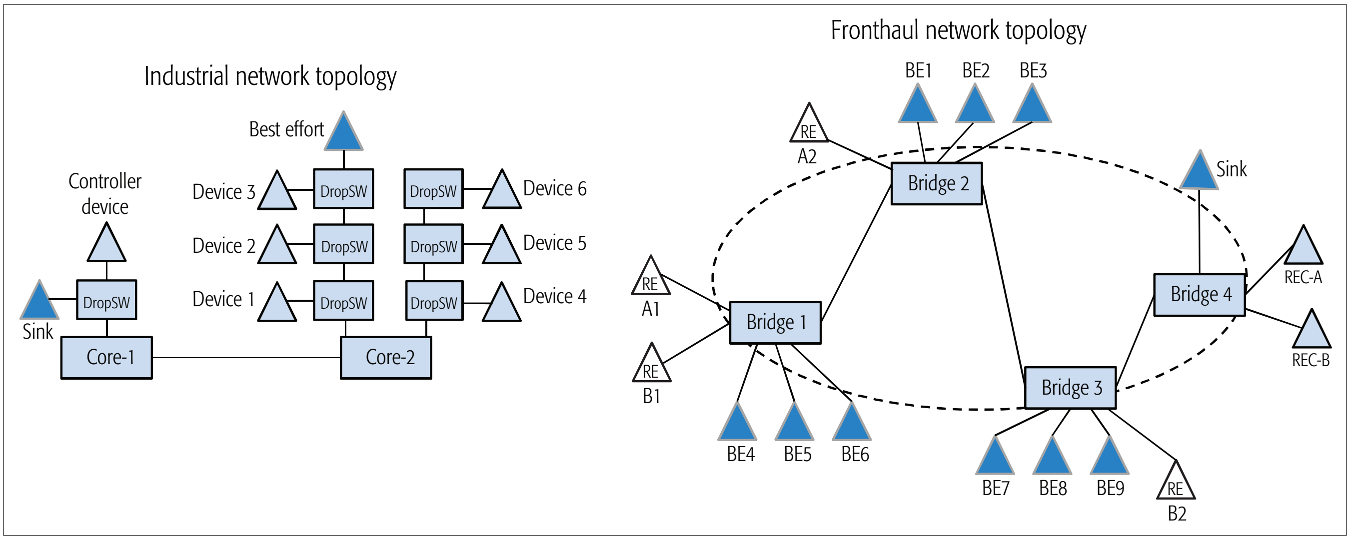

Figure 2. Simulated network topologies.

mitted (maximum 672 ns latency at the beginning of the preamble of the preemptable frame).

Figure 1 shows how a 124-octet frame is split into two 64-octet fragments. Non-final fragments use only 60 octets of the original frame content since they are extended with a new 4-octet checksum. Frames shorter than 124 octets cannot be split into valid fragments; thus, the worst case preemption latency equals the transmission delay of a 123-octet frame (e.g., 1080 ns on a 1 Gp/s link).

Time Gating and Preemption Combined

Time gating and frame preemption may be used separately, but it is possible to combine them for a greater effect. One way is to simply turn on frame preemption in addition to time gating. Its effectiveness, however, depends heavily on the time gate configuration, as detailed later.

The other way of combining frame preemption with time gating is through the Hold/Release mechanism, introduced by IEEE 802.1Qbu and IEEE 802.1Qbv. Shortly before opening the time gate of the express queue, a Hold signal is generated, which triggers the preemption of the ongoing preemptable transmission if there is such transmission in progress and forbids retrieving another preemptable frame from the queue. When the express gate closes, it emits a Release signal, which lifts the ban on processing preemptable traffic.

This way, there is a time gap before the protected transmission window for the express frames, called a guard band. The delay between the Hold signal and the opening of the express gate ensures that by the time the express frames arrive, the lower-priority traffic cannot delay the transmission. Obviously, to use a guard band, one must only enable time gating for the express queues.

Simulation Tool Implementing the TSN Features

For our simulation study we wanted to use a tool that is capable of frame-level analysis of the network traffic. We decided to use the INET model library of the OMNeT++ discrete event simulation framework [9], because it offers versatile simulation management and result evaluation tools, and the simulation models are easy to extend with new functionality. We implemented the time gating and frame preemption features in the full-duplex Ethernet medium access control (MAC) model, and added support for IEEE 802.1p priority queueing in the Ethernet transmit queue.

In the following sections the analyzed effects and design recommendations are all based on our simulation results, and we summarize the significant issues that have been found without the technical details. For this article we selected simulation results where network topologies and traffic parameters are dimensioned such that they illustrate interesting phenomena that arise when using TSN. The switching delays are constant, and clock synchronization inaccuracies are not modeled, because we wanted to illustrate the operational effects independent of the synchronization issues. We use the packet delay variation (PDV) metric to express variation in latency, calculated as the difference between the highest and lowest latency values [10]. Our results showing 0 PDV correspond to the ideal case, but in reality the PDV can be close to, but never reach, zero.

Industrial Networking Scenarios

In the following we discuss the applicability of TSN mechanisms using simulation-based experiments, providing insight for engineers designing such networks/services. In our first use case, that of industrial applications, different network topologies can be chosen (e.g., bus, tree, or ring), depending on the requirements. The typically used Ethernet link rates were 10/100 Mb/s a decade ago; nowadays, 100 Mb/s is generally used; and an upgrade to 1 Gb/s is expected in the near future.

In industrial field control, the real-time applications require strict timing of the control cycles between the controller and the controlled devices. This control traffic consists of small frames sent periodically with constant bit rate (CBR). The cycle

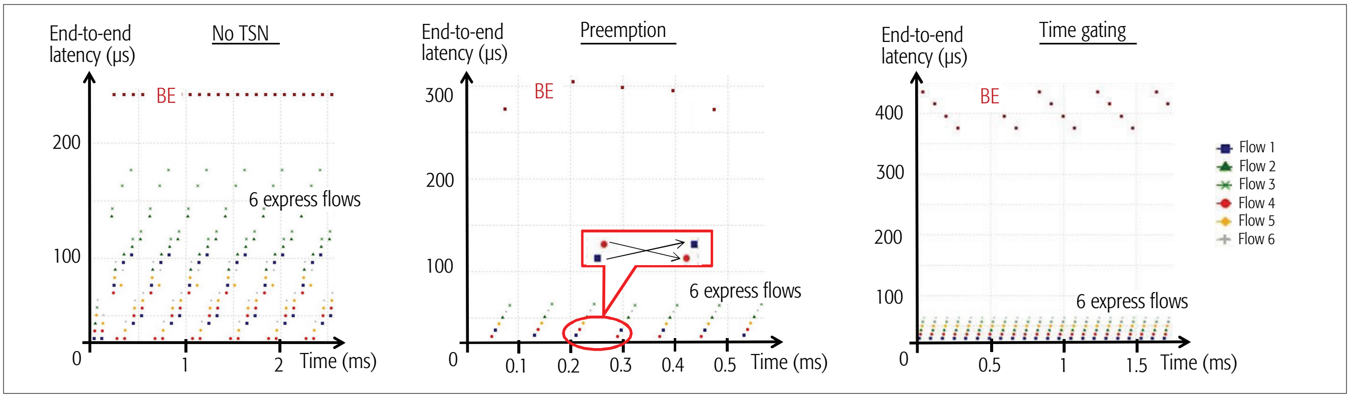

Figure 3. End-to-end latencies in industrial networks.

time of the control loops is in the sub-100 μs (motion control and robotics [11]) range, and it might be as low as a few microseconds [12] (factory automation, multi-axis CNC machines [13]).

Not only latency, but also PDV should be kept under control. PDV results in low-quality work due to irregular tool movement, and many machines are programmed to halt under such conditions [11].

To support our discussions in this section, we use the industrial bridged network topology shown in Fig. 2. There are core switches interconnected with each other (they may form a ring for redundancy), and there are drop switches connected to the core switches via cascaded buses. The end devices are connected to the drop switches one by one. All devices generate time-critical (express) traffic with 64-octet frames being controlled by one controller device. Best effort (BE) flow between the source-sink pair represents non-critical traffic, which may be preempted.

Figure 3 shows the end-to-end latencies of frames from all six express flows (plotted with separate colors) and that of the BE flow as a function of the simulation time, comparing three cases: not using TSN features, using only preemption, and using only time gating.

Preemption

Without any TSN feature used (see the left part of Fig. 3), the latency values may exceed 100 μs, and the PDV value (shown in Table 1) is also in this range.

Larger (450 octets) BE frames over the slow 100 Mb/s links have long transmission delays, and if an express frame is delayed by a BE frame, the end-to-end latency becomes very high. Note that the PDV suffered by a flow depends on its relative position to the BE source, too.

To avoid the large delays caused by the interfering BE frames, first we applied the preemption mechanism. This reduces the latency, and ensures a predictable pattern to the express frames. Nevertheless, the PDV is still 7.68 μs, a value that is an order of magnitude larger than the typical preemption latency on a 100 Mb/s link. The reason for this is that we scheduled the frames to arrive close to each other. Thus, the preemption latency is large enough to "shift" an express frame behind another express frame from a different control flow. The PDV is increased with the transmission delay of the other express frame, as highlighted in the middle part of Fig. 3.

No TSN (μs)	Preemption (μs)		Time gating (μs)
	Original frame timing (densely spaced)	Modified frame timing (sparsely spaced)	
119.3	7.68	1.36	0

Table 1. Worst PDVs in industrial networks.

Time Gating

Time gating can completely separate the transmission of frames belonging to different priority classes (see the chart on the right of Fig. 3 and Table 1). If time gating is applied to both the express and BE traffic, they can never disturb each other. It is also possible to protect two express flows from each other this way, if they are configured to be in different priority classes.

Time gating can have a negative impact on time-critical flows when the express frames miss their time slots. Missing a time slot and waiting for the next one not only amplifies the latency, but it can also push more frames out of their slots, thus creating an avalanche (unless the excess frames are evicted from the queue). Only careful planning and increased safety margins can avoid this. The duration while the time gate is open should equal at least the transmission delay of the express frame and the maximum PDV it may suffer.

An effective combination, especially when the Hold/Release mechanism is enabled, is to apply time gating only to the express traffic and to use preemption on the BE frames. A further advantage of this choice is that it does not impose a limit on the size of the preemptable frames: if the time between two consecutive open periods is not enough to transmit the preemptable frame, that frame will be fragmented.

On the other hand, time gating only the BE traffic places fewer restrictions on the express traffic, and the possibility of missed time slots is avoided.

Spacing of the Express Frames

Depending on which TSN features are used, the spacing between the express frames (i.e., the time between the arrivals of two consecutive frames) greatly influences the achieved QoS. Spacing of the express frames is the responsibility of the sending end stations.

First, looking at the case when *only preemption* is used, the PDV is influenced by the spacing between the express frames. If frames of the

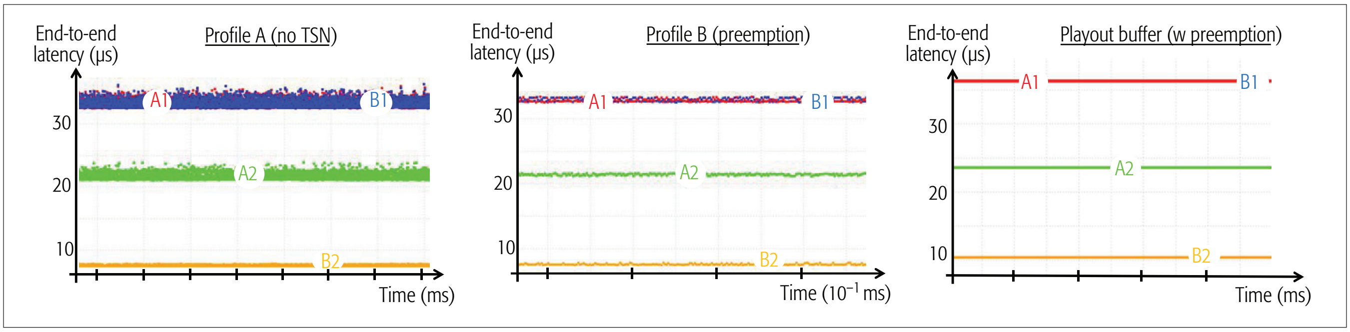

Figure 4. End-to-end latencies in fronthaul networks (BE frame latencies are not shown).

express flows are densely spaced (i.e., frames arriving closely one after another), the preemption latency causes them to interfere with each other, as mentioned above. If the frames are sparsely spaced, the PDVs are minimized. We modified the timing of the frames to keep distance between them; thus, the PDV equaled the preemption latency (Table 1).

Next, we investigate the case when time gating protects the express traffic from the BE traffic. This solution is effective only if the "closed" intervals for the BE traffic are calculated by taking into account the uncertainties of express frame arrivals; thereby, the gates are closed for longer intervals, and there will be less bandwidth available for the BE flows.

If the express frames are positioned to arrive close to each other (i.e., in packet trains to the switches), the gates can remain open for the BE traffic for a long continuous period within a cycle. Note that in this case BE frames cannot interfere with the express frames, so the larger PDV problem caused by the preemption latency is not an issue.

If the express frames do not arrive in train, multiple open-close gate operations will fragment the time period remaining for BE frames, which may decrease the maximum transferable frame size.

Depending on which TSN feature is used, different strategies should be followed in the scheduling of the express frames. Only using preemption comes with the advantage of bypassing both the synchronization issues and the complex task of scheduling the time gate control lists. This might be useful if the traffic is very dynamic in nature and frequent reconfigurations would be required. An operator might only use time gating in its network to achieve sub-microsecond-level precise timing. Deployments using preemption, including the combined use of time gating and preemption, might not be feasible due to the higher implementation cost of preemption.

In summary, the application of TSN tools in an industrial network may decrease the end-to-end latency from 100 μs to less than 10 μs, bringing it to the region where ultra-low-latency real-time control applications become feasible. Our results have also shown that, depending on the traffic pattern and the particular TSN tools applied, differences of more than 5 μs can be observed in the end-to-end latency. Note that the propagation delay of a 100-m-long link is 0.5 μs; thus, the lower latency of a good configuration can be used to either deploy applications with shorter control loops or cover a larger area.

Fronthaul Scenarios

Fronthaul networks, our second use case, deploy Common Public Radio Interface (CPRI) [14] to carry packetized radio data and therefore have specific requirements for synchronization, latency, and jitter. In this article we build on our previous simulation experiments investigating the implications of the TSN features on carrying fronthaul traffic over a bridged network, done as part of the discussions related to draft the IEEE 802.1CM standard [15]. The standard proposes two options: Profile A does not use TSN, while Profile B uses preemption.

Fronthaul networks, compared to typical industrial networks, cover longer distances but have fewer hops, higher link speeds (minimum 10 Gb/s), and larger frames. In our simulations we used 1.228 Gb/s CPRI flows, which means 983 Mb/s without the 8b/10b line coding that is unnecessary when being transported over Ethernet. After adding the Ethernet overhead, we get 1006.6 Mb/s when using 1500-octet frames and 1101 Mb/s for 300-octet frames.

In ideal conditions, the CPRI frames are sent at fixed, regular intervals. Multiple effects can increase the PDV of a flow, even if it is carried over an empty network. First, there are synchronization errors between the transport and the radio network. Then the remote radio sites cannot always perfectly synchronize their clocks; thus, there is a traffic source timing inaccuracy. Finally, the switching delays of the switches can also have small variations in time.

While in industrial networking the operator of the network has full control to schedule individual devices, REs in the fronthaul send the frames synchronously, and the transport network operator cannot modify this. Thus, express frames from different flows can arrive at a switch around the same time, and different engineering solutions have to be applied to handle the traffic timing.

We use the network topology shown in Fig. 2 with 1-km-long links of 10 Gb/s capacity, spanning over 4 hops, the bridging delay being 1.5 μs. While the real-life networks might have longer links and higher link rates, this topology is suitable for highlighting the particularities of TSN-based fronthaul networks. Frames are generated by four REs, controlled by REC-A and REC-B, respectively. There are 9 background flow sources, with BE service only. In the followings we focus on the results obtained with the CPRI data split into 300-octet packets, and use the results for 1500 octets only to explain the effect of frame size selection.

Results of selected simulation experiments are shown in Fig. 4, used to discuss the particularities of Profile A, Profile B, and a variant of Profile B.

Preemption

Profile A uses no TSN mechanisms, only priority queuing. In this case, if the express frame arrives slightly later than a BE frame, it must wait for the transmission of the BE frame. Due to this increased latency, it may interfere with other express frames (normally it would not) later along the path, thus suffering very large PDVs. The resulting latency values are plotted on the left side of Fig. 4, while their values and the PDVs are shown in Table 2.

Flows A1 and B1 have similar delays and PDVs as they share their paths, while flow B2 has a much shorter path. Therefore, we present only the results for flows A1 and A2 in Table 2, and reflect the worst case conditions in the network. Note that the timescale of the plots in Fig. 4 is different from the plots in Fig. 3 to better show our findings for the fronthaul traffic.

As one would expect, preemption (Profile B) can drastically reduce the PDV when there is interfering BE traffic, without needing time synchronization in the network (see the chart in the middle of Fig. 4 and Table 2). However, the latency variation cannot be completely eliminated with preemption alone.

Playout Buffer

The PDV in the above scenario can be mitigated by the application of a playout buffer (also called de-jittering buffer) at the receiving end. Usually, the proper setup of the buffer parameters requires further calibration, but in our scenarios the ideal arrival sequence, including the minimal end-to-end latency and timing information, is readily available.

Implementation of the playout buffer requires the exact control of the serving rate, which we achieved by shaping the traffic at the output port of the last bridge. We obtained 0 PDV, as shown in Fig. 4. Per REC playout buffer is not enough to achieve 0 PDV if express frames may leave a bridge inside the network in varying order due to interfering frames from different flows arriving more or less at the same time to the same switch. In this case, per flow playout buffer can solve this issue; however, it is not scalable for large numbers of flows.

Delayed Service with Time Gating

Even if interfering frames arrive at a bridge, eventually resulting in frame order variation on the output port, deterministic operation can still be achieved by smart time gating configuration. Express frames can be deterministically delayed on the previous bridges to avoid the interference on the next bridge, yielding deterministic output order (delayed service). When time gates are configured for delayed service, packets are held back with a closed gate and are released a little later, introducing artificial delay.

Delayed service can be beneficial even when there are no race conditions between express flows, because it can also eliminate various effects that cause PDV (e.g., switching delay variation, synchronization errors). The delayed service solution can eliminate PDV by withholding the express frames for the maximum duration of their accumulated PDV. The arriving frames that suffer shorter delays have to wait until the gate opens; thus, the delay up to that hop is homogenized. To put it another way, the deterministic delays introduced by gating can be traded for PDV reduction on a per-hop basis, not necessarily zeroing it out (Table 2).

Flow	Min delay (ns)	Max delay (ns)	PDV (ns)
Profile A (No TSN) – 300 octets			
A1	32,621.7	36,475.2	3853.5
A2	21,189.0	23,922.0	2733.0
Profile B (Preemption) – 300 octets			
A1	32,612.9	33,329.1	716.2
A2	21,169.2	21,592.5	423.3
Playout buffer – 300 octets			
A1	36,239.3	36,239.3	0
A2	24,139.3	24,139.3	0
Delayed Service (Time Gating) – 300 octets			
A1	32,611.0	32,805.6	194.6
A2	21,165.9	21,331.5	165.6
Profile A (No TSN) – 1500 octets			
A1	38,358.0	44,535.4	6177.4
A2	27,397.1	31,135.7	3738.6
Profile B (Preemption) – 1500 octets			
A1	38,362.4	41,035.8	2673.4
A2	27,397.2	28,795.1	1397.9
Playout buffer – 1500 octets			
A1	51,739.4	51,739.4	0
A2	38,339.4	38,339.4	0

Table 2. PDVs in fronthaul networks.

Fronthaul Network Size

In case of fronthaul networks it is an important design aspect that the express frame size influences the maximum network diameter.

There is a maximum end-to-end latency allowed by the CPRI standard. This latency is composed of the switching delays and transmission delays at every hop, the queuing delays, and the propagation delays. The RE-to-REC distance determines the propagation delay, which is 5 μs over a 1 km link. The smaller the other delay components are, the longer the links can be. The switching delay, link rates, and hop count also depend on the network devices and topology. The transmission delay and queuing delay components, though, depend on the frame size. Note that a larger frame size not only increases the transmission delay, but also the worst case value of the queuing delay.

We illustrate the effect of the frame-size on the fronthaul diameter by comparing the latency difference between the 300-octet- and 1500-octet-long CPRI packets, as shown in Table

Express frames can be deterministically delayed on the previous bridges to avoid the interference on the next bridge, yielding deterministic output order (Delayed Service). When time gates are configured for Delayed Service, packets are held back with a closed gate, and are released a little later, introducing artificial delay.

Industrial and fronthaul networks will certainly benefit from using time-sensitive networking, since they need to guarantee ultra-low latencies. Nevertheless, the design of the transport network, including the configuration of the TSN features, requires special care.

2. For example, for Profile B the difference in worst case latency is 7.7 μs; thus, smaller frames allow 1.5 km longer diameter.

Conclusions

Industrial and fronthaul networks will certainly benefit from using time-sensitive networking, since they need to guarantee ultra-low latencies. Nevertheless, the design of the transport network, including the configuration of the TSN features, requires special care. While frame preemption protects express traffic against low-priority traffic, time gating is required to achieve truly deterministic guarantees. The combination of these two features increases the efficiency of the solution.

Most of the TSN features require modifying the MAC layer; thus, their implementation on chip is a complex task that also impacts their cost. Nevertheless, once network chip vendors enter the market with products supporting the discussed TSN features, it will make the operation of TSN-enhanced standard Ethernet solutions simpler and more cost-effective.

The features discussed in this article, and generally TSN enhanced Ethernet, are expected to gain growing attention with the advent of 5G networks, since 5G will both demand ultra-low-latency transport solutions and enable applications requiring strong QoS guarantees.

Acknowledgment

The authors would like to thank the valuable comments received from János Farkas and Balázs Varga from Ericsson, and the help of their colleagues József Bíró, Árpád Péter Nagy, Norbert Bella, and István Moldován.

References

[1] Time-Sensitive Networking Task Group description; http://www.ieee802.org/1/pages/tsn.html, 2017.

[2] N. Finn, “Deterministic Networking Architecture,” Internet draft, 2017.

[3] ONC Ylmaz, “Analysis of Ultra-Reliable and Low-Latency 5G Communication for a Factory Automation Use Case,” *IEEE ICC 2015 — Wksp. 5G & Beyond*, June 2015.

[4] IEEE 802.1Qbv, “Enhancements for Scheduled Traffic,” 2015.

[5] IEEE 802.1Q-2014, “Bridges and Bridged Networks,” 2014.

[6] IEEE 802.1AS-Rev, “Timing and Synchronization for Time-Sensitive Applications,” 2017.

[7] IEEE 802.1Qbu, “Frame Preemption,” 2015.

[8] IEEE 802.3br, “Interspersing Express Traffic,” 2016.

[9] OMNeT++ v. 4.6 User Manual; https://omnetpp.org, 2015.

[10] A. Morton and B. Claise, “Packet Delay Variation Applicability Statement,“ IETF RFC 5481, 2009.

[11] G. Guerrini, “Virtualization Helps CNC Machines Consolidate Real-Time Processing,” *Electronic Design*, Apr. 2011.

[12] T. Mauer, “How to Add Industrial Ethernet to Computer Numeric Control (CNC) Router Machine,” Texas Instruments online training material, Apr. 2017.

[13] K. Klaver, “Enhance Mold Precision by Going Five-Axis,” *MoldMaking Technology*, Jan. 2013.

[14] CPRI Spec.; http://www.cpri.info/spec.html, 2017.

[15] IEEE Std. P802.1CM, “Time Sensitive Networking for Fronthaul,” 2018.

Biographies

Csaba Simon (simon@tmit.bme.hu) is a software engineer. He earned his Ph.D. in 2012 from Budapest University of Technology and Economics (BME) and currently works as an assistant professor in the Department of Telecommunication and Media Informatics at the same university. His research area includes the topics of future Internet, 5G networks, and cloud systems. He has participated in numerous national and international projects in the fields of network resource management, mobility management, and smart content delivery.

Markosz Maliosz received his M.Sc. (1998) and Ph.D. (2006) degrees in computer science in the field of infocommunication systems from BME. He is an associate professor in the Department of Telecommunication and Media Informatics, BME. His research interests include virtual, cloud, and sensor networking along with optimization techniques. He has participated in numerous national and international projects in the fields of network resource management, multimedia, and smart content delivery.

Miklós Máté received his MSc (2007) degree in computer science in the field of infocommunication systems from BME and currently is preparing to defend his Ph.D. dissertation at the same university. He is a research engineer in the High-Speed Networks Laboratory (HSNLab) in the Department of Telecommunication and Media Informatics, BME. His research interests include intelligent transportation systems and distributed networks.