Received February 24, 2019, accepted March 28, 2019, date of publication April 3, 2019, date of current version April 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2908613

Performance Comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS)

AHMED NASRALLAH[1], AKHILESH S. THYAGATURU[1], ZIYAD ALHARBI[1], CUIXIANG WANG[2], XING SHAO[2], MARTIN REISSLEIN[1], (Fellow, IEEE), AND HESHAM ELBAKOURY[3]

[1]School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287-5706, USA
[2]Department of Internet of Things Engineering, Yancheng Institute of Technology, Yancheng 224051, China
[3]Futurewei Technologies, Inc., Santa Clara, CA 95050, USA

Corresponding author: Martin Reisslein (reisslein@asu.edu)

The work of C. Wang was supported by the Overseas Training Program of the Yancheng Institute of Technology. The work of X. Shao was supported by the Jiangsu Provincial Government Scholarship for Overseas Learning.

ABSTRACT The IEEE 802.1 time sensitive networking working group has recently standardized the time aware shaper (TAS). The TAS provides deterministic latency guarantees but requires tight time synchronization in all network switches. This paper thoroughly evaluates the mean and maximum packet delays and packet losses of the TAS for a typical industrial control ring network for random (sporadic) and for periodic traffic. We propose and evaluate adaptive bandwidth sharing and adaptive slotted window mechanisms to make TAS adaptive to traffic fluctuations. This paper further evaluates the asynchronous traffic shaper (ATS), which has been proposed to provide low latency network service without the need for time synchronization in network nodes. Our evaluations indicate that TAS with proper configurations, e.g., accurate and precise gating schedules, generally achieves the specified latency bounds for both sporadic and periodic traffic. In contrast, ATS performs relatively well for sporadic traffic; but struggles for moderate to high loads of periodic traffic.

INDEX TERMS Asynchronous traffic shaper (ATS), packet delay, throughput, time-sensitive networking (TSN), time aware shaper (TAS), ultra-low latency.

I. INTRODUCTION

A. MOTIVATION

A wide range of communication network applications, including industrial network applications [1]–[3] and wireless network applications [2], [4]–[10], require ultra-low latency (ULL) on the order of milliseconds or less [11]–[14]. Ethernet technology has been widely adopted as link-layer connectivity standard in communication networks as a result of Ethernet's openness and cost effectiveness. However, traditional Ethernet has been designed to provide high link utilization so as to achieve maximum end-to-end throughput for best effort services [15], [16]. While best effort services provide high link utilization, and simple implementation, the end-to-end delay cannot be guaranteed. Hence, Ethernet is not well suited for applications that require deterministic end-to-end delay, such as industrial and automation control, professional audio/video production, and automotive control. The deterministic and low latencies required by these applications resulted in the development of semi-proprietary technologies, such as Time-Triggered Ethernet (TTEthernet), EtherCAT, and FlexRay [17] that limit interoperability and interconnectivity.

To address deterministic latency requirements and ULL requirements, the IEEE 802.1 Time-Sensitive Networking (TSN) Task Group (TG) has defined a suite of standards that extend Ethernet technology. The IEEE 802.1 TSN standardization extends the standard Ethernet services with additional features that provide deterministic guarantees, robustness, as well as integrated diagnostics and management services. These standards outline and define new mechanisms that enable distributed synchronized real-time systems using standard Ethernet technologies, allowing the convergence of high-priority low-latency scheduled traffic (ST) and standard

The associate editor coordinating the review of this manuscript and approving it for publication was Rongbo Zhu.

best effort (BE) Ethernet traffic on the same network. In this paper, we focus on the IEEE 802.1Qbv [18] Enhancements to Scheduled Traffic, i.e., the Time-Aware Shaper (TAS), and the IEEE 802.1Qcr [19] Asynchronous Traffic Shaper (ATS). TAS defines a mechanism for time-driven control and scheduling of data frames; whereas, ATS defines a mechanism for assigning eligibility times to received frames according to the ATS state machine.

B. CONTRIBUTIONS

We comprehensively evaluate the performance of TAS and ATS in terms of mean and maximum packet delays and packet loss ratios. In particular, we make the following contributions:

i) We conduct a rigorous evaluation of the TAS standard mechanisms with respect to a comprehensive set of parameters to assess performance characteristics and to reveal limitations. More specifically, we investigate the impact of the TAS shaper parameters, e.g., the ST to BE traffic gating proportions and traffic loads.

ii) We design and evaluate a novel TAS adaptive bandwidth sharing (ABS) mechanism to enhance link utilization. Moreover, we design and evaluate an adaptive control of the traffic gating proportions, namely the adaptive slotted window (ASW) mechanism, to provide low delays for scheduled traffic even for high network traffic loads. We examine the impact of the combination of these two adaptive mechanisms on ST and BE traffic.

iii) We compare the TAS and ATS shapers to evaluate whether ATS is capable of achieving similar performance as TAS in industrial networks with periodic and sporadic data transfer tasks. Based on the traffic and network models, we infer several characteristics that determine the advantages of using ATS and/or TAS.

C. ORGANIZATION

This article is organized as follows. Section I-D contrasts related work from our study. Section II provides background on the IEEE 802.1 TSN standardization. Section III explains the timing analysis and algorithms for TAS, adaptive TAS, and ATS, including their respective state machines. Section III-A.3 introduces the novel adaptive bandwidth sharing (ABS) and adaptive slotted window (ASW) mechanisms designed to overcome the limitation of standard TAS. Section IV-A introduces the simulation network model and traffic models. Section IV-B presents the evaluation results for the TSN standard and adaptive TAS, while Section IV-C presents the ATS results. Section V concludes the paper.

D. RELATED WORK

Generally, two main approaches are undertaken when designing real-time Ethernet based packet switched networks: *i*) synchronous time-triggered based medium access control (e.g., TAS), and *ii*) asynchronous event-triggered approach (e.g., ATS).

The TSN standardization, and in particular the realization of the time-driven TAS, involves strict synchronized time requirements [20], similar to Time Triggered Ethernet (TTEthernet) [21], to ensure accurate deterministic service. Although TTEthernet shares Time Division Multiplexing (TDM) similarities with TAS, TAS operates on predefined traffic classes and not on individual frames as TTEthernet does. Additionally, TAS, as part of the IEEE 802.1 TSN standardization, can be coupled with several TSN standards, e.g., IEEE 802.1Qci [22] (PSFP), IEEE 802.1Qch [23] (CQF), to add fine-grained control and management to ST flows so as to provide highly precise guaranteed deterministic behaviors.

FTT-Ethernet [24], [25] is a master/multi-slave (MS) protocol that leverages the design principles of switched Ethernet and provides time-triggered scheduling to guarantee timeliness. FTT-Ethernet enforces dynamic quality of service (QoS) and admission control to ensure guaranteed bandwidth and bounded network induced latencies, specializing in Controller Area Network (CAN) field networks. However, the FTT-Ethernet protocol has not gained widespread adoption due to lack of standardizations/certifications and timeliness guarantees. Meyer *et al.* [26] have presented a network scenario to analyze the impact of time-triggered scheduling on audio video bridging (AVB) class A traffic in automotive scale networks. Specifically, Meyer *et al.* [26] analyzed the impact of time-triggered communication on competing traffic on the same infrastructure.

The IEEE TSN TG has published a series of standards that govern stream/flow shapers within 802.1 bridges/switches, and in particular the TAS mechanism. A first analysis of TAS and related TSN shapers has been presented by Thangamuthu *et al.* [27]. Thangamuthu *et al.* performed a comprehensive timing analysis of the TSN shapers with emphasis on the end-to-end delays for high-priority Control Data Traffic (CDT). Thangamuthu *et al.* argued that TAS achieves the lowest latencies and jitter. Similar to Meyer *et al.*'s automotive topology, Thangamuthu *et al.* evaluated three proposed traffic shapers, namely TAS, Peristaltic Shaper [23], and Burst Limiting Shaper, on a small automotive topology. In contrast, in this study, we examine industrial control networks with varied periodic and sporadic traffic for synchronous (TAS) and asynchronous (ATS) shapers and introduce novel adaptation mechanisms for TAS.

Thiele *et al.* [28] have compared TAS against the TSN Peristaltic Shaper [23] and standard 802.1Q Ethernet, i.e., scheduling purely based on the frame priority. Thiele *et al.* provided significant insight into the operation and timing analysis of switched Ethernet using TAS (and the Peristaltic Shaper), including delay analysis for several blocking effects and delay analysis of TSN and non-TSN streams. Park *et al.* [29], Farzaneh and Knoll [30], and Maxim and Song [31] have evaluated the performance of TAS in automotive in-vehicle networks focusing on automotive use-cases. Nsaibi *et al.* [32] have evaluated TAS in the

specific context of a Sercos *III* network. In contrast, we perform a comprehensive evaluation of TAS in the context of general industrial control networks and compare TAS with the emerging Asynchronous Traffic Shaper (ATS).

For completeness, we note that several studies have examined a range of specific aspects of operating TSN networks. A theoretical worst-case delay analysis of TAS has been conducted in [33]. Dürr and Nayak [34] have derived an offline scheduler that given periodic time-triggered Ethernet frames can optimally schedule and reduce gate-driver entries with minimized end-to-end delays. Craciunas *et al.* [35], [36] presented a scheduling method to compute static schedules for TAS using Satisfiability Modulo Theory (SMT) and Optimization Modulo Theory (OMT) solvers. Craciunas *et al.* identified key functional constraints affecting the behavior of TSN networks which are used to set a generalized configuration of parameters for real-time ST streams. Lander *et al.* [37] have developed a heuristic algorithm that reconfigures TAS switches according to runtime network conditions, while related routing and scheduling schemes have been further studied in [38]. Feasible schedules are produced and forwarded using a configuration agent (composed of a Centralized User Configuration (CUC) and Centralized Network Configuration (CNC)). The model of Lander *et al*. emphasizes appearing and disappearing flows in a fog computing platform that takes into account the flows' properties and possible routes.

Nayak *et al.* [39] have explored how routing impacts the TAS scheduling. Furthermore, Nayak *et al.* [40] have employed Software Defined Networking (SDN) to incrementally add new TAS flows while preserving the QoS of existing TAS flows. Moreover, several recent studies, e.g. [41]–[47], have explored time-triggered scheduling for 5G fronthaul networks. Furthermore, in-car Ethernet communications have gained significant traction, specifically for automotive applications, such as multimedia/infotainment and Advanced Driver Assistance Systems (ADASs) [48].

The ATS standard is still in the draft state. At this point, only few studies have examined ATS in the TSN context. Zhou *et al.* [49] have examined two ATS alternatives, namely *i*) the Urgency Based Scheduler (UBS), and *ii*) Paternoster policing and scheduling. For UBS, two main interleaved algorithms have been presented, in particular, frame-by-frame leaky bucket and token based leaky bucket. In this paper, we focus mainly on token based ATS since the ATS draft standard follows a token bucket based algorithm. The simulations in Zhou *et al.* [49] consider only one-hop transmission of sporadic traffic, with emphasis on the comparison of Paternoster and UBS. In contrast, this study considers a more realistic networking scenario with multi-hop transmissions of sporadic and periodic traffic. Emphasizing the need to prevent burstiness cascades in TSN and extending the interleaved shapers/regulators analyzed in [50], Mohammadpour *et al.* [51] have mathematically analyzed credit based shaper (CBS) and ATS service curves for AVB traffic classes as well as bounds on the CBS and ATS response times and traffic backlogs.

II. BACKGROUND: IEEE 802.1 TIME SENSITIVE NETWORKING (TSN)

A. IEEE 802.1Qbv: TIME AWARE SHAPER (TAS)

TAS operates in a similar manner as TTEthernet [21], i.e., based on time-triggered scheduling. While TTEthernet implements time-based scheduling of individual frames or flows, the TAS schedule is based on predefined traffic classes. Scheduling based on traffic classes, as opposed to individual frames or flows, provides better scalability. Since traffic classes are defined according to the priority code point (PCP) values of the Virtual Local Area Network (VLAN) ID (VID) tag of 802.1Q frames, only applying 802.1Qbv TAS cannot enable fined-grained identification and control on the level of individual streams or flows. Additional mechanisms, such as Per-Stream Filtering and Policing (PSFP) [22] and Frame Replication and Elimination for Reliability (FRER) [52], allowing identification of frames based on the Stream ID and overriding of the traffic class encoded in the PCP code, would be necessary to achieve the same level of per-flow QoS as in TTEthernet [53].

TAS schedules high-priority critical traffic streams in reserved time-triggered windows. In order to prevent lower priority traffic, e.g., best effort (BE) traffic, from interfering with the scheduled traffic (ST) transmissions, ST windows are preceded by a so-called guard band. TAS needs to have all time-triggered windows synchronized, i.e., all bridges from sender to receiver must be synchronized in time, usually through the 802.1AS time reference [18, Secs. 8.6.8.4 and 8.6.8.4.10]. TAS utilizes a gate driver mechanism that opens and closes according to a known and agreed upon time schedule for each port in a bridge. In particular, the Gate Control List (GCL) contains Gate Control Entries (GCEs), i.e., a sequence of 1's or 0's that represent whether a queue is eligible to transmit or not. The frames of a given egress queue are eligible for transmission according to the GCL, which is synchronized in time.

Frames are transmitted according to the GCEs in the GCL and transmission selection decisions. Each individual software queue has its own transmission selection algorithm, e.g., strict priority queuing. Overall, the IEEE 802.1Qbv transmission selection transmits a frame from a given queue with an open gate if: (*i*) The queue contains a frame ready for transmission, (*ii*) higher priority traffic class queues with an open gate do *not* have a frame to transmit, and (*iii*) the frame transmission can be completed before the gate closes for the given queue. Note that these transmission selection conditions ensure that low-priority BE traffic is allowed to *start* transmission only if the transmission will *be completed* by the start of the window for high-priority ST traffic. Thus, this transmission selection effectively enforces a ''guard band'' to prevent low-priority traffic from interfering with high-priority traffic.

B. IEEE 802.1Qcr: ASYNCHRONOUS TRAFFIC SHAPER (ATS)

While TAS performs well in imposing traffic determinism, the stringent timing requirements, in particular the high required precision levels of the timing synchronization across the TSN domain, increase complexity and threaten the reliability of the TSN network domain if any timing misalignment occurs. Furthermore, several synchronization challenges, e.g., skew or drift in timing signal frames, clock inaccuracy, and lost timing frames, can lead to inaccuracies downstream from the synchronized master clock in the TSN domain. As the network scale increases, so does the synchronization complexity.

As an alternative to TAS, the Asynchronous Traffic Shaping (ATS) (IEEE 802.1Qcr [19]) imposes similar traffic determinism without the need for strict timing synchronization. Initially, ATS was proposed based on research by Specht and Samii [54] on an Urgency Based Scheduler (UBS) that operates according to two approaches: *i*) Length-Rate Quotient (LRQ) and *ii*) Token Based Emulation (TBE). In this paper, we focus on the token bucket based approach due to the draft standard's similar approach to ATS.

The token bucket based ATS approach achieves QoS through asynchronously operating sub-shapers. Talkers (transmitters) can decide when to send as long as they comply with the prescribed rate limits. The ATS sub-shapers regulate the traffic at every hop at the granularity of an individual stream (or flow) or an aggregate of multiple streams (or flows). Each switch operating UBS includes a number of shaped queue instances associated with a number of shaper groups that are controlled internally through the use of Internal Priority Values (IPVs) and interleaved regulators, whereby each group is scheduled using an independent local clock. UBS thus effectively implements hierarchical per-hop shaping.

III. TIMING MECHANISMS AND TRANSMISSION ALGORITHMS

This section provides more detail on the TAS and ATS mechanisms, specifies the main TAS and ATS parameters and state variables, and gives the main TAS and ATS algorithms. Figs. 1 and 2 illustrate the finite state machine operation, including the main operating states and event transitions, of TAS and ATS, respectively.

A. TAS

1) TIME AWARE SCHEDULING

TAS considers two main traffic types, namely high-priority scheduled traffic (ST) and low-priority best effort traffic (BE). ST is buffered in the ST queue, and BE traffic is buffered in the BE queue within switches, i.e., TAS implements frame priority isolation by traffic class. TAS divides up the transmission opportunities so as to deterministically satisfy the ST QoS bounds. TAS ensures that the ST delay is bounded and protects ST from any cross-traffic interference.

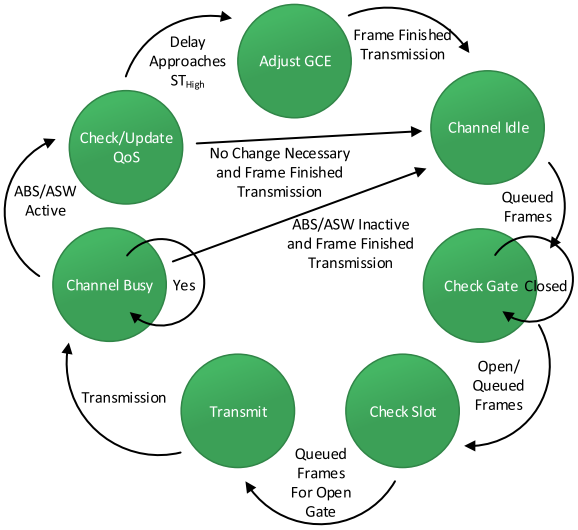


FIGURE 1. Time Aware Shaper (TAS) state machine illustrating the states and transitional operations for both standard and adaptive TAS.

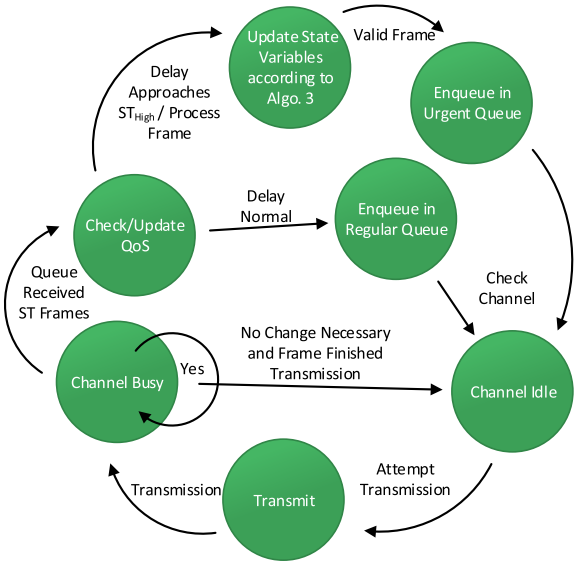


FIGURE 2. Asynchronous Traffic Shaping (ATS) state machine illustrating the states and transitional operations.

TAS switches shift the gate status to open/close depending on the cyclic GCL composed of several GCEs. The GCL is programmed to follow a strict TDMA approach that depends on a global synchronized time (in our simulation, that time is *simTime()*).

Fig. 3 shows a timing diagram where the guard band (although not explicitly present) is inherent in the TAS implementation. Since the size of each packet is known, the transmission delay can be calculated. If the current ST packet that is awaiting transmission on a idle output channel has a total transmission time less than the time duration until the start of the BE slot time, then TAS will send the ST packet. Otherwise, the ST packet is scheduled for the next GCL cycle. Therefore, if an ST packet arrives at the beginning of a BE slot, then it has to wait at least until the beginning of

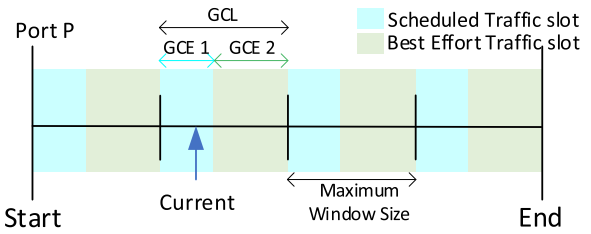


FIGURE 3. Illustration of Time Aware Shaper (TAS) time line: Each GCL cycle has a duration equal to the maximum window time, i.e., the cycle time (CT). Gates are opened and closed according to the GCEs within a give GCL cycle.

the next ST slot. While the ST to BE proportions are fixed in Fig. 3, we consider dynamically changing the proportion ratio according to the current runtime delay experienced at the receiver in Section III-A.3.

2) SCHEDULING ALGORITHM

1) *Maximum Window Time [Cycle Time (CT)]*: The Maximum Window Time, which is also referred to as cycle time (CT), specifies the basic time period for the GCL to repeat. The total gate times (windows) that can be allocated within one CT to the various traffic classes must sum to less than or equal to the CT.
2) *Slotted Windows*: Slotted windows define the timed transmission windows given to each traffic class on a given switch. Initially, these values are statically predefined. However, with a centralized management method and accurate traffic characterization, it is possible to dynamically define the slotted window times at runtime, i.e., achieve TAS reconfiguration. In our implementation, the Priority Ratio ST^R indicates the proportion of the cycle time that is allocated to the slotted window for the ST class. The BE class is allocated the leftover proportion of the CT.
3) *Gating Mechanism*: The gating mechanism is the primary mechanism that given a time-based signal (timer) blocks or unblocks a queue from transmission. In our implementation, we calculate the current window slot based on the global simulation time in OMNeT++ (Lines 6–8 in Algorithm 1) and according to the calculated slot, check if the current simulation time belongs to the ST or BE slot (Lines 9 and 17), effectively blocking any transmission from consideration depending on the current global synchronized simulation time.
4) *Queue Management*: Queue management determines the removal of frames within queues as well as the internal queue structure and management. For our implementation, each queue (total of two traffic class queues for each egress port) is implemented as a *cPacketQueue* container class using the strict priority structure. Packets are inserted into the back of the ST or BE queue according to the packet traffic class (ST or BE). FIFO is considered within a given queue.

Our initial simulations evaluate TAS without bandwidth sharing or changing the gating ratio at each switch port

Algorithm 1 Time Aware Shaper (TAS) algorithm applied to each switch output (egress) port for two traffic classes, namely ST and BE. The TAS standard specifies fixed static bandwidth allocations and windows. We introduce novel adaptive bandwidth sharing (ABS) and adaptive slotted windows (ASW), see Section III-A.3. (The "!" indicates a negation, e.g., "!isEmpty(Queue)" means that the queue is *not* empty.)

1: $Port_P$: Considered Egress Port
2: ST^R: Given ST Gating Proportion for GCE
3: CT: Given Cycle Time (max. window time) for GCL
4: **procedure** TAS($Port_P$)
5: **if** $Port_P$ is idle **then**
6: $S_i \leftarrow simTime_{current}/CT$
7: $ST_{slot} \leftarrow floor(S_i) * CT + ST^R * CT$
8: $BE_{slot} \leftarrow ceil(S_i) * CT$
9: **if** $simTime_{current} \leq ST_{slot}$ **then**
10: **if** !isEmpty(ST_Q) **then**
11: **if** $\frac{Pkt_{length}^{current}}{Channel_{capacity}} \leq ST_{slot}$ **then**
12: send($Packet_{ST}$, $Port_P$)
13: **else**
14: Schedule(ST_{slot})
15: **end if**
16: **end if**
17: **else**
18: **if** !isEmpty(BE_Q) **then**
19: **if** $\frac{Pkt_{length}^{current}}{Channel_{capacity}} \leq BE_{slot}$ **then**
20: send($Packet_{BE}$, $Port_P$)
21: **else**
22: Schedule(BE_{slot})
23: **end if**
24: **end if**
25: **end if**
26: **else**
27: Schedule($simTime_{current} + \frac{Pkt_{length}^{transmitted}}{Channel_{capacity}}$)
28: **end if**
29: **return**
30: **end procedure**

with OMNeT++ according to Algorithm 1, while TAS with adaptive bandwidth sharing (ABS) and adaptive slotted windows (ASW) are implemented using Algorithm 2. In Algorithm 2, the ST^R is updated whenever an update message is received at a specific egress port on the switch. Our ASW implementation (see Section III-A.3) follows a distributed approach of propagating an end-point message to allow the ST slot to expand or shrink depending on the perceived runtime delay at the end-point (e.g., sink). The message is propagated in the reverse path of the affected flows.

3) MOTIVATION AND SPECIFICATION FOR AN ADAPTIVE TAS

Evaluations of the standard TAS (which are presented in detail in Section IV) indicated an overall shortcoming of

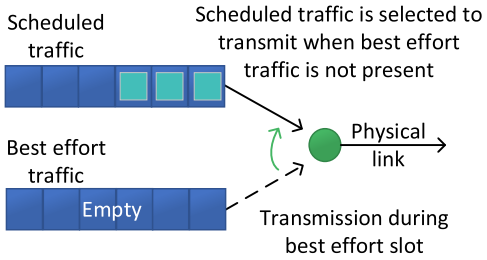


FIGURE 4. Time Aware Shaper (TAS) Adaptive Bandwidth Sharing (ABS) illustration.

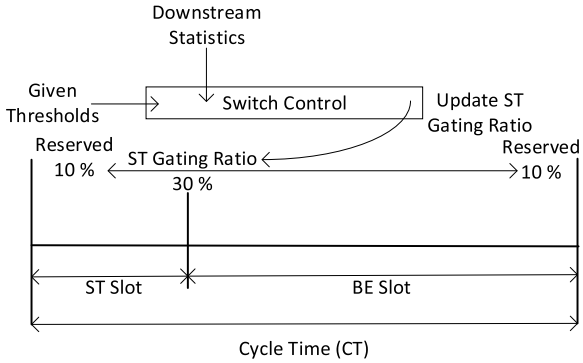


FIGURE 5. Time Aware Shaper (TAS) Adaptive Slotted Windows (ASW) illustration.

standard TAS in that the gating ratio between ST and BE traffic was a limiter that can increase or decrease the ST delay while inversely impacting the BE delay. Additionally, link utilization had been observed to be very low due to strict slotted windows that when not configured correctly can deteriorate the ST and BE QoS. Therefore, we propose to mitigate such limitations by introducing a controllable adaptive bandwidth sharing (ABS) and a dynamic adaptive slotted window (ASW) mechanism. ABS and ASW are ST centric in the sense that they favor ST traffic over BE traffic following Algorithm 2.

a: ADAPTIVE BANDWIDTH SHARING (ABS) SPECIFICATION

To enable high utilization, we propose adaptive bandwidth sharing (ABS) of the ST and BE slot times to blocked queues if a given transmission opportunity would otherwise go unused. That is, ABS temporarily shares the bandwidth of the current slot time. ABS is specified in Algorithm 2, specifically in Lines 26 and 36.

As shown in Fig. 4, the transmission opportunity reserved for BE traffic (due to the BE slot timer) can be temporarily shared with ST traffic. When the timer is set to BE or ST, the non-empty (ST or BE) waiting queue can be selected for transmission if the reserved (BE or ST) queue is empty. While this operation may contradict the TAS operation of preventing cross-interference, the idea behind ABS is to further reduce the ST delays while keeping BE delays relatively low, effectively mitigating BE traffic starvation.

b: ADAPTIVE SLOTTED WINDOWS (ASW) SPECIFICATION

We propose an Adaptive Slotted Window (ASW) mechanism that shifts the ST to BE gating ratio according to runtime network statistics. This adaptive approach should be able to cope with bursts of ST traffic while temporarily sacrificing the BE QoS. For the ASW modification of the standard TAS mechanism, we feed back receiver runtime network statistics to the upstream switches. The upstream statistics reporting can be implemented over listener (receiving node) to multiple talkers (sending nodes) trees, analogous to the recently proposed listener microstream-interleaving [55] in the context of TSN cyclic queueing and forwarding (CQF) [23] through updates of the traffic specifications.

As illustrated in Fig. 5, the upstream switch determines whether it is necessary to expand or reduce the slotted window according to a predefined delay threshold for a particular traffic class. If the current runtime delay approaches the threshold, then we expand the slot for the traffic class in question, and vice versa.

Specifically, we update the gating ratio through a step size of $\Delta = 0.1$. The TSN standardization and recent literature typically set the maximum ST traffic delay bound to $ST^R_{\max} = 0.1$ ms for a maximum of 5 switch hops. For the lower threshold, we select half, i.e., $ST^R_{\min} = 0.05$ ms, to avoid frequent updates of the ST gating ratio. Additionally, we reserve 10% of the CT for each traffic class to avoid severe congestion when the network is highly loaded, as specified in Algorithm 2, specifically in Lines 7–13. Future research could examine the impact of the granularity of the step size Δ.

B. ATS

ATS shapers assign eligibility times to frames belonging to specific streams which are then used for traffic regulation by the ATS transmission selection algorithm [19, Sec. 8.6.8.5] without adhering to global network time synchronization. According to the draft standard [19], each bridge provides an ATS Shaper Instance Table with parameters and variables of up to *MaxShaperInstances* independent ATS shaper instances, an ATS Shaper Group Instance Table with parameters and variables of up to *MaxShaperGroupInstances* independent ATS shaper group instances, and an ATS Port Parameter Table with parameters and variables shared by all ATS shaper instances associated with a reception port. To evaluate pure ATS without added complexity or confounding protocol mechanisms, we simulated ATS in isolation without Per-Stream Filtering and Policing (PSFP) [22].

Each ATS shaper instance assigns eligibility times to the associated frames, and discards frames in rare situations. The underlying operations are performed by an ATS shaper state machine [19, Sec. 8.6.11] associated with an ATS shaper instance which is built following the token bucket algorithm. This state machine updates the associated bucket empty time and group eligibility time state variables based on the $TokenRate_{size}$ parameter, the $TokenBurst_{size}$ parameter, the $Max_{ResidenceTime}$ parameter, the frame arrival times, and

Algorithm 2 Adaptive Time Aware Shaper (Adaptive TAS) algorithm applied to each output port of the switch for two traffic classes, namely ST and BE. The Adaptive Bandwidth Sharing (ABS) mechanism is specified in Lines 26 and 36. The Adaptive Slotted Window (ASW) mechanism is specified in Lines 7–13

1: $Port_P$: Considered Egress Port
2: ST_R: ST Gating Ratio initialized to some value
3: ST_{High} Given ST maximum threshold
4: ST_{Low} Given ST minimum threshold
5: Δ Given gating ratio step size
6: CT: Given Cycle Time (maximum window time) for GCL
7: **procedure** Calculate $ST^R(ST_R)$
8: **if** $Delay_{Sink}^{current} \geq ST_{High}$ **AND** $ST^R + \Delta \leq 0.9$ **then**
9: $ST^R = ST^R + \Delta$
10: **else if** $Delay_{Sink}^{current} \leq ST_{Low}$ **AND** $ST^R - \Delta \geq 0.1$ **then**
11: $ST^R = ST^R - \Delta$
12: **end if**
13: **end procedure**
14: **procedure** TAS($Port_P$)
15: **if** $Port_P$ is idle **then**
16: $S_i \leftarrow simTime_{current}/CT$
17: $ST_{slot} \leftarrow floor(S_i) * CT + ST^R * CT$
18: $BE_{slot} \leftarrow ceil(S_i) * CT$
19: **if** $simTime_{current} \leq ST_{slot}$ **then**
20: **if** !isEmpty(ST_Q) **then**
21: **if** $\frac{Pkt_{length}^{current}}{Channel_{capacity}} \leq ST_{slot}$ **then**
22: send($Packet_{ST}$, $Port_P$)
23: **else**
24: Schedule(ST_{slot})
25: **end if**
26: **else if** !isEm.(BE_Q) **AND** $\frac{Pkt_{len.}^{cur.}}{Ch.cap.} \leq ST_{sl.}$ **then**
27: send($Packet_{BE}$, $Port_P$)
28: **end if**
29: **else**
30: **if** !isEmpty(BE_Q) **then**
31: **if** $\frac{Pkt_{length}^{current}}{Channel_{capacity}} \leq BE_{slot}$ **then**
32: send($Packet_{BE}$, $Port_P$)
33: **else**
34: Schedule(BE_{slot})
35: **end if**
36: **else if** !isEm.(ST_Q) **AND** $\frac{Pkt_{len.}^{cur.}}{Ch.cap.} \leq BE_{sl.}$ **then**
37: send($Packet_{ST}$, $Port_P$)
38: **end if**
39: **end if**
40: **else**
41: Schedule($simTime_{current} + \frac{Pkt_{length}^{transmitted}}{Channel_{capacity}}$)
42: **end if**
43: **return**
44: **end procedure**

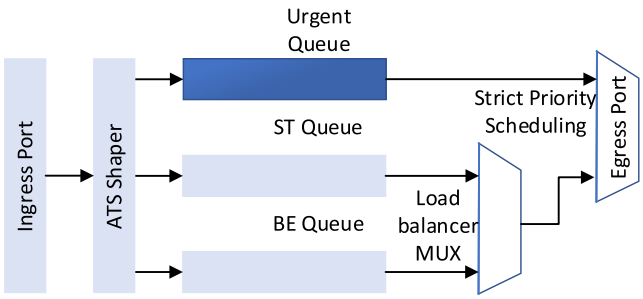


FIGURE 6. Illustration of Asynchronous Traffic Shaper (ATS) bridge operation: The ATS shaper at the ingress determines whether to regulate/shape ST traffic flows utilizing the urgent queue. The traditional ST and BE queues follow a fair multiplexed transmission scheme allocating fair transmission times to both ST and BE.

the total frame length. If an ATS shaper instance discards a frame, the discard frame counter of the associated port is increased. For bridges with ATS support, and without support for Enhanced Scheduled Traffic (i.e., TAS) and PSFP, the ATS traffic stream gates are permanently open and only used for Internal Priority Value (IPV) assignment, i.e., traditional queue arbitration and management. Fig. 6 illustrates the high level overview of the ATS model implemented in OMNeT++. All received frames need to pass the ATS shaper module before being admitted to the queues. The ATS shaper directs some traffic to the urgent queue according to the ATS state variables, eligibility time (of a given considered frame), and the current QoS experienced at the current hop, i.e., per-hop shaping. After shaping is performed, strict-priority scheduling is used to grant channel access to frames at the egress queue. The regular ST and BE queues are multiplexed at the egress to allow fair sharing of the channel, i.e., to prevent starvation of BE traffic. All queues are FIFO queues.

Algorithm 3 specifies the general concept for implementing the ATS shaper state machine in an ingress bridge port. A local clock *ATS_ShaperClock* is used to determine arrival times of frames at a given ingress port. Upon the arrival of an ST frame to the switch ingress port, the current time stamp is tagged to the incoming ST frame. Similarly, the departure times from the egress ports are tagged to the ST frames so that the elapsed times in the various switches on the path to the destination sink can be tracked [19, Sec. 8.6.11.2.1.]. The accumulated elapsed time is compared against the ST_{high} threshold [19]. If the elapsed time is less than $0.8 \times ST_{high}$, then the frame is enqueued in the ST queue. If the elapsed time is greater than $0.8 \times ST_{high}$, then the ATS_ProcessFrame procedure is invoked for the ST frame. The ATS_ProcessFrame procedure returns the assigned eligibility time to the ST frame and queues the ST frame to the urgent queue, or designates the frame for discard. In our evaluations, ST frames that would be discarded by the standard Alg. 3, Line 19, are enqueued in the ST queue. Received BE frames are directly enqueued in the BE queue.

Algorithm 3 The Asynchronous Traffic Shaper (ATS) defined in IEEE 802.1Qcr Draft 0.5 [19] computes the ST frame Eligibility Time (ET), assigns the ET to each ST Frame, and updates the ATS state machine variables in each ingress port

1: $TokenRate_{size}$ Given token rate
2: $TokenBurst_{size}$ Given token burst size
3: $Max_{ResidenceTime}$ Given max residence time
4: $Bucket_{EmptyToFullDuration} = \frac{TokenBurst_{size}}{TokenRate_{size}}$ Given total bucket recovery duration
5: **procedure** ATS_ProcessFrame($Frame_{Received}$)
6: $Bucket_{LengthRecoveryDuration} = \frac{Frame_{size}}{TokenRate_{size}}$
7: $ATS_{ShaperEligibilityTime} = Bucket_{EmptyTime} + Bucket_{LengthRecoveryDuration}$
8: $Bucket_{FullTime} = Bucket_{EmptyTime} + Bucket_{EmptyToFullDuration}$
9: $Frame_{EligibilityTime} = MAX(Frame^{ArrivalTime}_{Received}, ATS_{GroupEligibilityTime}, ATS_{ShaperEligibilityTime})$
10: **if** $Frame_{EligibilityTime} \leq Frame^{ArrivalTime}_{Received} + Max_{ResidenceTime}$ **then**
11: $ATS_{GroupEligibilityTime} = Frame_{EligibilityTime}$
12: **if** $Frame_{EligibilityTime} < Bucket_{FullTime}$ **then**
13: $Bucket_{EmptyTime} = ATS_{ShaperEligibilityTime}$
14: **else**
15: $Bucket_{EmptyTime} = ATS_{ShaperEligibilityTime} + Frame_{EligibilityTime} - Bucket_{FullTime}$
16: **end if**
17: $Queue_{Urgent} \leftarrow Frame_{Received}$
18: **else**
19: Discard_Frame($Frame_{Received}$)
20: **end if**
21: **end procedure**

The urgent queue follows a token bucket algorithm to regulate the traffic according to the following state variables used in Alg. 3.

1) $TokenBurst_{size}$ – The maximum token capacity of the token bucket, in bits, for an ATS shaper instance.
2) $TokenRate_{size}$ – The rate at which the token bucket is refilled with tokens until the maximum token capacity $TokenBurst_{size}$ is reached, in bits per second.
3) $Bucket_{EmptyTime}$ – A state variable that contains the most recent time instant when the token bucket of the ATS shaper instance was empty, in seconds. At initialization, the number of tokens in the token bucket is set to the $TokenBurst_{size}$.
4) $Bucket_{EmptyToFullDuration}$ – The time duration required to accumulate a number of tokens equivalent to the $TokenBurst_{size}$, in seconds.
5) $Bucket_{FullTime}$ – The most recent time instant when the number of tokens in the token bucket is equivalent to the $TokenBurst_{size}$, in seconds.
6) $Bucket_{LengthRecoveryDuration}$ – The duration required to accumulate a number of tokens equivalent to the frame length, i.e., the length of the currently considered frame, in seconds.
7) $Frame_{EligibilityTime}$ – The eligibility time (time to send) of the frame, without considering the device-internal forwarding processing delays.
8) $Max_{ResidenceTime}$ – This parameter limits the duration for which frames can reside in a bridge, in nanoseconds.
9) $ATS_{GroupEligibilityTime}$ – A state variable that contains the most recent $Frame_{EligibilityTime}$ from the previous frame, as processed by any ATS shaper instance in the same ATS shaper group, in seconds.
10) $ATS_{ShaperEligibilityTime}$ – The earliest time when there are enough tokens in the bucket to transmit the packet.

The assigned eligibility time (used by the ATS transmission selection algorithm) is calculated by measuring the variation (offset) between the local ATS shaper clock and the ATS transmission selection clock, and the forwarding processing delay within a switch. Our evaluations assume that the processing delay and the time offset between the ATS shaper local clock and the ATS transmission clock are negligible. Therefore, the assigned eligibility time is governed by the token bucket process in the ATS state machine (Lines 13 and 15 in Alg. 3). The assigned *EligibilityTime* calculated by the ATS_ProcessFrame procedure is agnostic to device internal parameters and the link characteristics.

A frame is eligible for transmission if the assigned eligibility time [19, Sec. 8.6.11.2.2] is less than or equal to the current time. The current time is determined by the Transmission Selection Clock, which is a local system clock. The Transmission Selection Clock determines the selectability time per frame, which is the time at which the arrival frame is queued [19, Sec. 8.6.6] and available for transmission selection. All frames that reach their selectability time are selected for transmission in ascending order of the assigned eligibility times. Frames with identical assigned eligibility times are selected according to the ordering requirement specified in [19, Sec. 8.6.6].

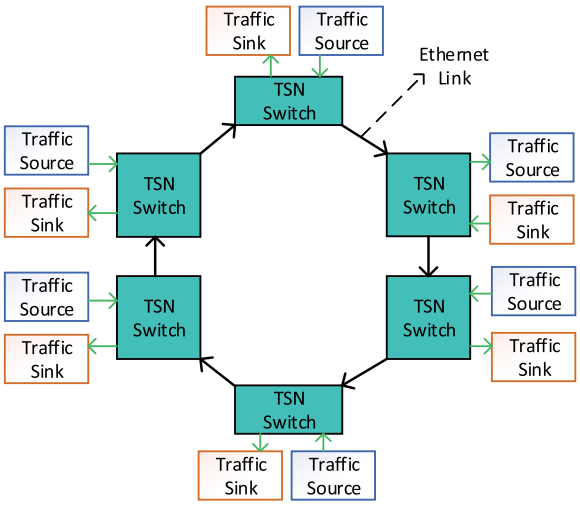


FIGURE 7. Industrial control loop topology [56]: Each source sends data in the clockwise direction traversing a varying number of hops around the ring.

IV. PERFORMANCE EVALUATION

A. SYSTEM OVERVIEW AND SIMULATION SETUP

This section explains the simulation setup, i.e., presents the considered industrial network topology and simulation scenarios. Throughout, we employ the OMNeT++ [57] simulation environment.

1) NETWORK MODEL

We consider a ring network topology. The ring topology is ubiquitous for industrial networks, which typically require ultra-low delay service. In particular, we consider a ring consisting of six switches, as shown in Fig. 7. The switch-to-switch links operate as full-duplex Ethernet links with a capacity (transmission bitrate) of $R = 1$ Gbps. One traffic source and one traffic sink are directly attached to each switch. The distance between two successive switches around the ring is 100 m, corresponding to a propagation delay of 0.5 μs. The switch egress port buffer size is set to 512 Kbyte (KB) for each traffic type, i.e., each switch egress port has a 512 KB buffer for ST traffic and a 512 KB buffer for BE traffic.

2) TRAFFIC MODEL

According to the International Electrotechnical Commission (IEC) and the IEEE TSN task group there is wide range of use cases for TSN [58]. These use cases can generate a wide variety of traffic, including isochronous (periodic) traffic, e.g., for control loops, as well as random (sporadic) traffic, e.g., from alarms or event monitors. In order to conduct a comprehensive evaluation, we consider sporadic traffic as well as periodic packet traffic.

We generate sporadic packet traffic according to independent Poisson processes with the same packet generation rate at each traffic source. Each traffic source independently randomly generates data packets of size 580 bytes. For the sporadic traffic model, both scheduled traffic (ST) and best effort traffic (BE) are generated according to independent Poisson processes. We remark that we employ the terminology "scheduled traffic (ST)" to indicate high-priority traffic that is to be transmitted in the high-priority ST traffic slots of the GCL cycles. In our sporadic traffic scenario, the scheduled traffic (ST) is generated at random times (i.e., asynchronously) in the traffic sources (i.e., it is not scheduled in advance); nevertheless, for consistence with the common TSN terminology, we refer to this sporadic high-priority traffic as "scheduled traffic (ST)". In particular, ST and BE traffic are generated as per the traffic proportions in Table 1, i.e., 20% of the generated packets are high-priority ST packets, and 80% of the generated packets are low-priority BE packets. The traffic intensity is characterized through the relative traffic load, i.e., the traffic intensity relative to the $R = 1$ Gbps link transmission bitrate. For instance, a load of $\rho_L = 0.5$ corresponds to a total bitrate of 0.5×1 Gbps injected into the network across the six source nodes, whereby each source node uniformly contributes one sixth of the total load.

TABLE 1. Traffic ratio scenarios (ST and BE traffic proportions relative to the total traffic load ρ_L) for sporadic (Poisson) traffic model, and corresponding TAS ST gating ratios ST^R, whereby the cycle time is set to $CT = 50$ μs.

	Traffic		Gate	
	ST	BE	ST	BE
Scenario 1	20%	80%	20%	80%
Scenario 2	20%	80%	30%	70%

We also consider a periodic (pre-planned) traffic model, with periodic high-priority ST and sporadic Poisson low-priority BE traffic. Each ST traffic source has a periodic traffic generation module that is synchronized to the cycle time structure of the switches. Thus, each traffic source generates a prescribed number of ST packets and injects them into the network right at the instant when the ST traffic slot starts at the switch that the traffic source is attached to. For periodic ST traffic, we consider 64 byte packets (which are typical for control data traffic (CDT)). Each traffic source injects a prescribed number of π, $\pi = 1, 2, 4$, or 8 CDT traffic packets (of 64 bytes each) at the beginning of each ST slot (i.e., once per cycle). Thus, the periodic traffic contributes a fixed ST bitrate of $\pi \times 64 \times 8$ bit $\times 6$ source nodes $/CT$. The corresponding ST intensity is obtained by dividing the ST bitrate by the link transmission bitrate R. In addition to the ST intensity, the network is loaded with a BE traffic intensity of ρ_L. For clarity, we report the delay performance results separately for different values of the number of periodic ST traffic packets π. That is, we show separate curves for $\pi = 1, 2, 4$, and 8. Each plot shows the performance as a function of the BE (background) traffic intensity ρ_L.

Each packet independently randomly travels a hop distance of one, two, three, four, or five switch-to-switch hops in the clockwise direction around the ring with probabilities 0.1, 0.1, 0.1, 0.3, and 0.4, respectively. Note that for the uniform load of multi-hop packet traffic injected at each source node in conjunction with the traffic routing in one

direction along the ring, the ring nodes experience uniform steady-state packet traffic loading. Thus, as the load level is increased, all ring nodes experience the same high steady-state traffic loading level, i.e., all ring nodes become essentially bottlenecks. Therefore, the performance at a bottleneck node with a prescribed high loading level in a different topology, e.g., in a mesh topology, is essentially equivalent to that of a ring bottleneck node with the same high loading level.

For a given evaluation scenario, we simulate 100 seconds of network operation, which corresponds to over 30 Million packets for a load of 2.0.

3) TAS AND ATS SETTINGS

In this paper, we report TAS results for a cycle time $CT = 50\ \mu$s, which is commonly considered for TSN studies. In additional evaluations that are not included due to space limitations, we conducted evaluations for $CT = 100\ \mu$s and found similar results as for $CT = 50\ \mu$s. Two ST to BE gate ratios are considered, see Table 1. The scenario 1 (S1) gate ratio matches the ST to BE traffic ratio. Scenario 2 (S2) gives ST traffic a 30% proportion of the gate times while the ST traffic is still only 20% of the total traffic.

The ATS parameters were set to $ST_{High} = 0.1$ ms and $ST_{Low} = 0.05$ ms, as well as $TokenRate_{size} = 128$ KB/s, $TokenBurst_{size} = 512$ KB, and $Max_{ResidenceTime} = 20\ \mu$s based on extensive empirical trials that sought to achieve low ST packet delays, while providing a reasonable overall packet traffic service.

B. TAS EVALUATION

1) SPORADIC ST SOURCES

We first compare standard TAS with the proposed adaptive TAS.

a: ADAPTIVE BANDWIDTH SHARING (ABS)

Fig 8(a) shows the average end-to-end packet delay for both standard TAS (Std) and adaptive TAS with ABS. We observe that for scenario 1 (S1) with equal ST/BE traffic and gating ratios, the standard TAS approach gives slightly higher delays for ST than for BE. This is mainly due to the relatively small ST windows for the considered ST/BE ratio of 20/80. ABS with its dynamic window utilization mitigates the effects of that small ST windows and reduces the mean ST delays to close to the BE delays. For the more typical TAS operating scenario 2 (S2) with a slightly higher ST/BE gating ratio than the ST/BE traffic ratio, we observe from Fig 8(a) that the ST mean packets delays are significantly lower than the BE delays. We also observe that ABS leaves the ST delay unchanged, while significantly reducing the BE delays.

Fig. 8(b) shows the maximum ST packet delays. We observe that for S2, the maximum packet delays are significantly lower than for S1. Nevertheless, these maximum ST packet delays are significantly higher than the typical

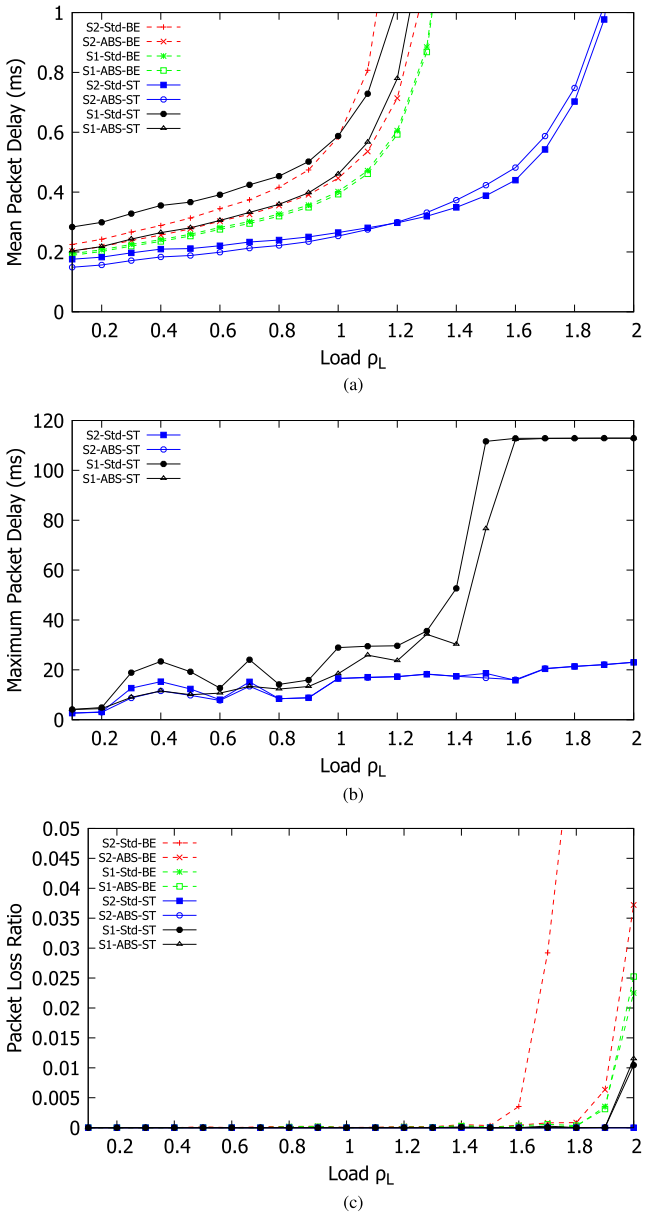


FIGURE 8. TAS with Adaptive Bandwidth Sharing (ABS) compared to standard TAS for sporadic ST sources. (a) Mean packet delay, (b) Maximum packet delay, and (c) Packet loss ratio.

ST delay targets on the order of a millisecond. This result indicates that sporadic (random) traffic can experience worst-case delays of ten or more milliseconds with standard TAS and TAS with ABS.

Fig. 8(c) shows the total packet loss ratios. We observe that ST and BE traffic experience no loss at low to moderate loads. We observed from Fig. 8(a) that in the S1 scenarios, the mean ST delays were higher than the corresponding mean BE delays. Now, we observe from Fig. 8(c) that for the S1 scenarios, the ST packet losses are lower than the BE traffic losses. ST traffic has smaller losses since ST traffic has the same buffer space (512 KB) available as BE traffic, but ST traffic has a four times smaller traffic volume than BE traffic (see Table 1). We also observe that for S2, the ST

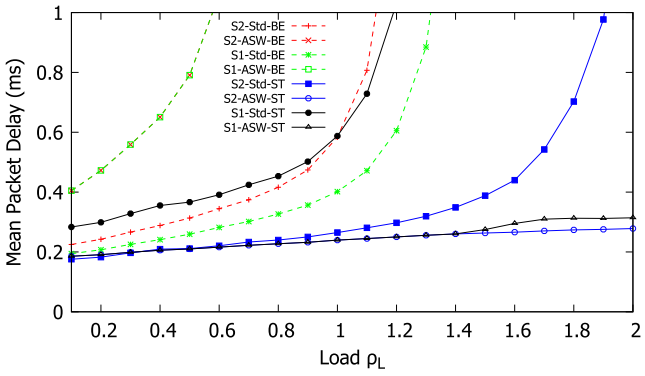


FIGURE 9. Mean packet delay for TAS with Adaptive Slotted Windows (ASW) compared to standard TAS for sporadic ST sources. (The curves for S1-ASW-BE and S2-ASW-BE are overlapping.)

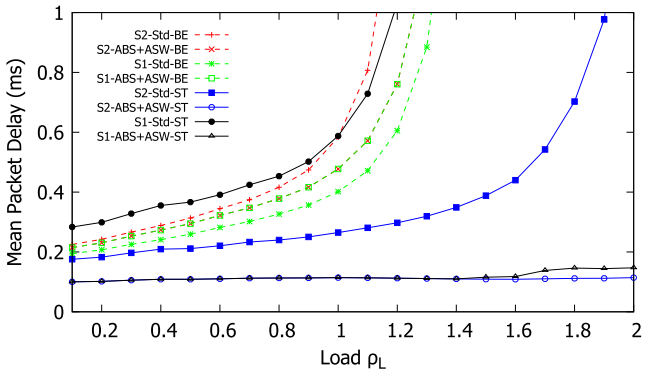


FIGURE 10. Mean packet delay for TAS with combined ASW and ABS compared to standard TAS for sporadic ST sources. (The S1-ABS+ASW-BE and S2-ABS+ASW-BE curves are overlapping.)

packet loss is consistently zero (for both standard TAS and for TAS with ABS); this is due to the overprovisioning of the gating ratio in favor of ST. Moreover, we observe for S2 that ABS reduces the BE packet loss compared to standard TAS.

b: ADAPTIVE SLOTTED WINDOWS (ASW)

Fig. 9 shows the end-to-end average ST and BE packet delays for TAS with ASW compared to the standard TAS. We observe for S1 with the initially equal ST/BE traffic ratios and gating ratios that ASW achieves substantial ST delay reductions compared to standard TAS; whereby the delay reductions are most pronounced at high loads. We also observe that the S1-BE delays are correspondingly increased by ASW. These delay results indicate that ASW effectively expands the ST window to consistently ensure low ST delays, even when the initial ST/BE gating ratio setting does not favor ST traffic. In particular, we observe from Fig. 9 that Scenarios 1 and 2, which differ in the initial gating ratio settings give essentially equivalent ASW delays. These equivalent delays are due to the continuous gating ratio updates of the proposed ASW mechanism, i.e., with ASW, the packet delays are over the long run independent of the initial ST/BE gating ratio since the ASW mechanism dynamically adapts the ST/BE gating ratio.

c: ABS & ASW COMBINED

Fig. 10 shows the mean ST and BE packet delays for TAS with the combined ABS and ASW in comparison to the standard TAS. We observe that similar to the TAS with ASW-only delays in Fig. 9, the combined ABS+ASW achieves substantial ST delay reductions compared to the standard TAS. Further comparisons of Figs. 9 and 10 indicate that the combined ABS+ASW reduces the mean delays from slightly above 0.2 ms for ASW-only to around 0.1 ms for ABS+ASW. Moreover, we observe from the comparison of Figs. 9 and 10 that the combined ABS+ASW substantially reduces the BE delays. For instance, for S1 with a load of $\rho_L = 0.4$, ASW-only gives a mean BE packet delay of approximately 0.65 ms in Fig. 9; whereas the combined ABS+ASW gives a corresponding mean BE packet delay of 0.27 ms in Fig. 10. Thus, the ABS+ASW combination can extract substantial additional delay reductions for both ST and BE packets through the dynamic ABS sharing across the ST and BE gating windows on top of the underlying ASW dynamic adaptation of the gating ratios.

In additional evaluations that are not included due to space constraints, we found that the S2 maximum ST frame delays for the combined ABS & ASW are below 4 ms for all load levels. Thus, the combined ABS & ASW achieves substantial reductions from the maximum ST frame delays of up to around 20 ms for standard TAS and TAS with ABS in Fig. 8. An interesting future work direction is to add frame preemption [59] to TAS with the combined ABS & ASW in order to further reduce the maximum (worst-case) ST packet delays.

2) PERIODIC ST SOURCES

Similar to the evaluation for sporadic (Poisson) ST traffic sources, we have compared the proposed adaptive TAS mechanisms to standard TAS for periodic ST traffic sources, as specified in Section IV-A.2. We have considered the periodic ST traffic injection rates $\pi = 1, 2, 4$, and 8 ST packets per CT. However, due to space constraints, we present only plots for $\pi = 4$ and 8; the plots for $\pi = 1$ and 2 are very similar to the plots for $\pi = 4$. Also, for brevity, we only present results for TAS with combined ABS & ASW.

Fig. 11 shows the average end-to-end ST and BE packet delays for $\pi = 4$ and 8 ST packets per CT. We observe from Fig. 11(a) that for $\pi = 4$, standard TAS consistently achieves very low mean ST delays below 0.01 ms (for both S1 and S2) for the entire range of BE traffic loads. In contrast, the mean ST delays for TAS with ABS & ASW (both for S1 and S2) increase nearly linearly with increasing BE traffic load until a load of around $\rho_L = 1.2$ and then flatten out around 0.06 ms. Turning to Fig. 11(b) for $\pi = 8$, we observe that the delays for TAS with ABS & ASW increase at a slightly steeper slope but flatten out at around the same level as for $\pi = 4$. On the other hand, the delays of standard TAS for $\pi = 8$ are 81 ms and 54 ms for S1 and S2 for the entire range of BE traffic loads. These results indicate that TAS with ABS & ASW

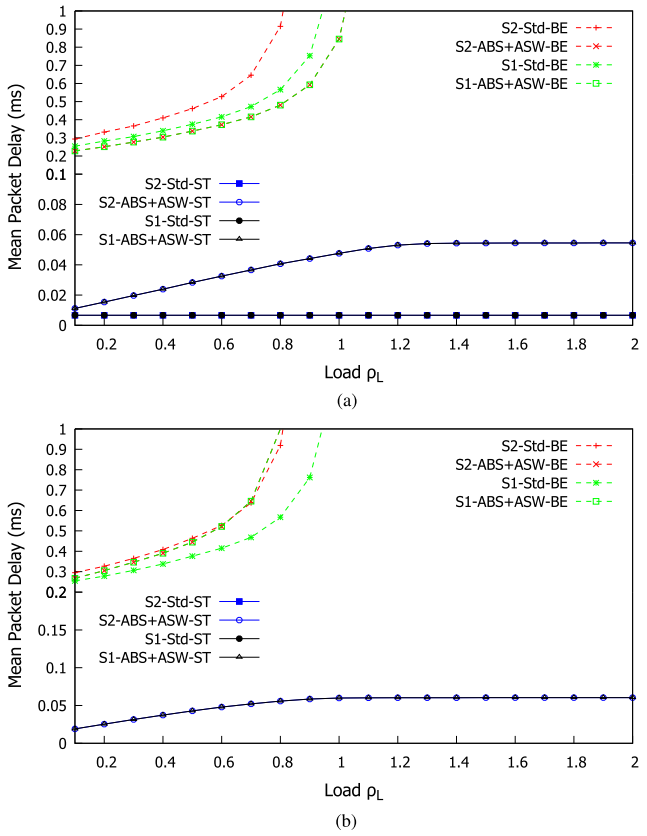


FIGURE 11. Mean packet delay for TAS with combined ASW and ABS compared to standard TAS for periodic ST sources that inject π ST packets per CT. (The S1-ABS+ASW-BE and S2-ABS+ASW-BE curves are overlapping.) (a) $\pi = 4$ ST packets/CT, (b) $\pi = 8$ ST packets/CT; the standard TAS ST delays are above the plotted range.

can provide robust low-delay service to ST traffic, even for relatively high loads of periodic ST traffic. Standard TAS with fixed parameter settings would require manual intervention to adjust to such high ST traffic loads. TAS with ABS & ASW automatically adjusts to high ST traffic loads and reduces the BE gate allocations so as to keep prioritizing ST traffic.

The mean packet delay results in Fig. 11 confirm that the ABS+ASW approach is independent of the initial setting of the gating ratio (S1 or S2 from Table 1 were considered). This result is expected since ASW reactively adapts the gating ratio; thus, the initial gating ratio setting becomes irrelevant. We proceed to consider ABS+ASW with only one initial setting in subsequent evaluations.

We also evaluated the maximum packet delays for $\pi = 4$ and 8 ST packets per CT. We found that the maximum ST packet delays were below 0.25 ms for $\pi = 4$. For $\pi = 8$ packets/CT, standard TAS S1 and S2 maximum delays were 108 ms and 71 ms, respectively, for all BE traffic loads ρ_L. In contrast, we observed that TAS with combined ABS & ASW consistently achieved maximum packet delays on the order of 0.2 ms.

Fig. 12 shows the packet losses for $\pi = 8$ ST packets/CT as a function of the BE load. We observe that TAS with ABS+AWS achieves zero ST packet losses throughout.

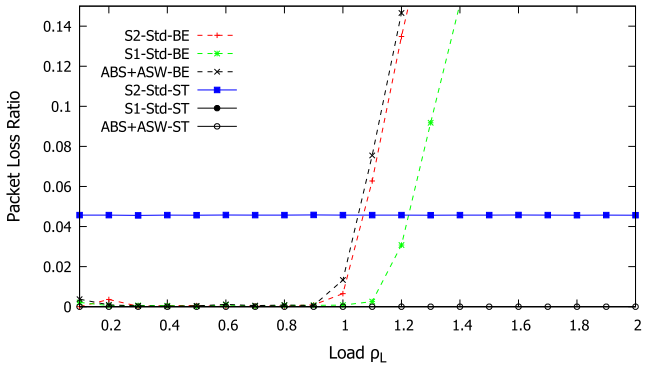


FIGURE 12. Packet loss ratio for TAS with combined ASW and ABS compared to standard TAS for periodic ST sources injecting $\pi =$ 8 packet/CT. For standard TAS, the ST packet loss for S1 is consistently at 0.37 (37%).

In contrast, standard TAS gives substantial ST packet losses, even for very low BE loads. The combined ABS+ASW drops BE traffic at approximately the same rate as standard TAS in S2.

3) SUMMARY OF PACKET DELAY VARIATION RESULTS

While the presented performance evaluation has focused on mean and maximum packet delays and losses, we have also evaluated packet delay variations (jitter). Generally, TAS strives for very short packet delays, accordingly, packet delay variations are expected to be small. In summary, we found for sporadic traffic that the ST delay variations (represented by the standard deviation of the packet delays) were on the order of 0.1 ms or less with adaptive TAS, while BE packets experienced delay variations up to 10 ms at high loads. Similarly, for the periodic traffic scenario, the ST packets had significantly smaller delay variations (on the order of 0.1 ms of less) than the BE packets (1 ms or higher for moderate to high loads). Overall, we also found that adaptive TAS gave lower packet delay variations than standard TAS.

C. ATS EVALUATION

For the evaluation of ATS (which does not have the concept of gating that TAS has), we consider the traffic proportions 20% of ST with 80% BE, as well as 30% ST with 70% BE.

1) SPORADIC ST SOURCES

Fig. 13 shows the mean and maximum ST and BE end-to-end delays as well as the packet loss ratio. Generally, we observe from Fig. 13 that ATS performs significantly better than standard TAS for sporadic traffic sources. In particular, we observe from Fig. 13(a) that for 20% ST traffic, ATS gives substantially lower ST delays than standard TAS; whereby the delay reduction with ATS is particularly pronounced compared to TAS S1.

We observe from Fig. 13(b) that ATS provides the same short maximum packet delays for both 20% and 30% ST traffic. In contrast, standard TAS gives relatively short maximum ST packet delays for S2, while the maximum ST packet delays for S1 shoot up to around 100 ms for moderately

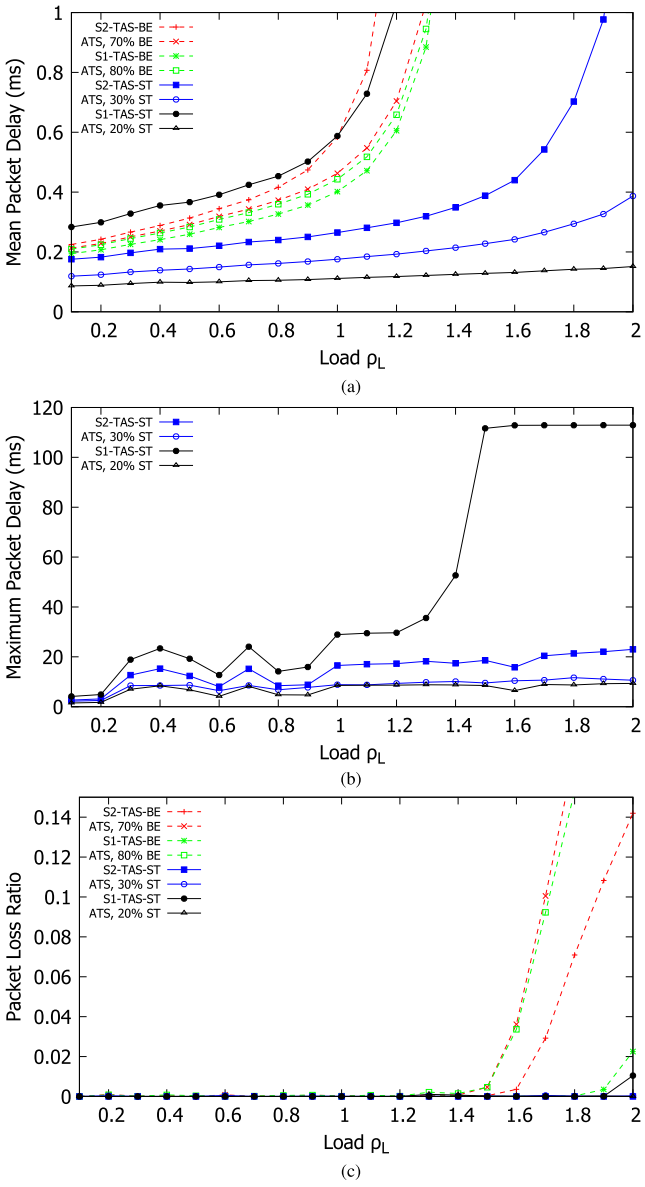


FIGURE 13. ATS compared with standard TAS for sporadic ST sources. (a) Mean packet delay, (b) Maximum packet delay, and (c) Packet loss ratio.

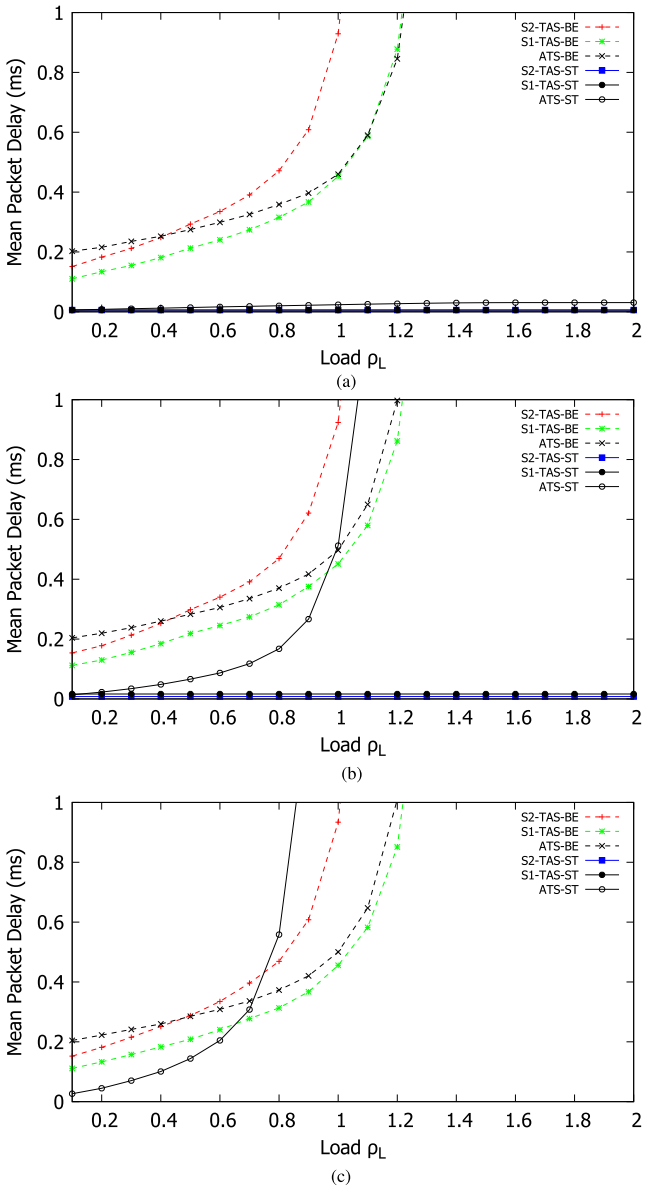


FIGURE 14. Mean packet delay for ATS compared with standard TAS for periodic ST sources. (a) $\pi = 2$ ST packets/CT, (b) $\pi = 4$ ST packets/CT, and (b) $\pi = 8$ ST packets/CT; S1 and S2 TAS ST delays are 71 ms and 53 ms, constant.

high loads. With ATS, each switch ingress queues the ST packets in the urgent queue if the runtime delay is close to the threshold (whereby we set the "close" parameter to 0.8 times the ST threshold). Within the urgent queue, the ATS algorithm follows the leaky bucket policy, ensuring consistent packet service.

We observe from Fig. 13(c) that ATS achieves nearly zero ST packet losses; the ATS ST losses are lower than the S1 TAS losses at very high loads. We also observe that ATS suffers from higher BE packet losses than TAS at high loads. This increased ATS BE packet loss is mainly due to increased delays at the BE queue and correspondingly higher probabilities of packet losses at the BE queue.

While the ATS simulation produced better results than TAS, the difficultly was mainly in setting and adjusting the configuration parameters for the ATS state variables such that the QoS for ST was guaranteed and the BE traffic was not starved. Shifting fixed static configuration management to dynamic variable configuration is needed to further enhance the granularity of ATS with respect to the number of flows and queue management schemes. Additionally, resource allocation and dropping rogue flows are needed to stop floods of large frames into the TSN domain so as to allow timely flows to proceed within the contract agreement.

2) PERIODIC ST SOURCES

Following the TAS evaluation for periodic ST sources, we similarly evaluate ATS for periodic ST sources. The ST traffic injection rates are set to $\pi = 2,\ 4$, and 8 ST packets per

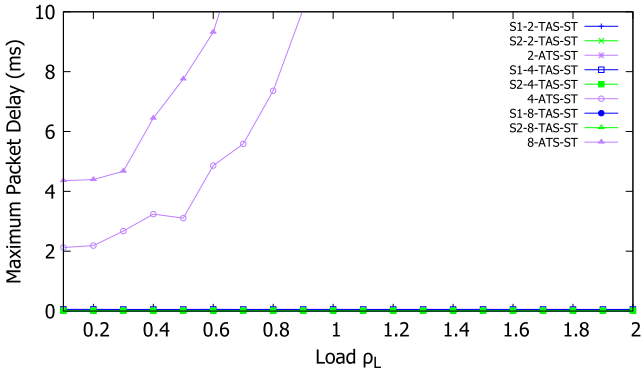


FIGURE 15. Maximum packet delay for ATS compared to standard TAS for periodic ST sources. The S1 and S2 TAS ST packet delays for $\pi = 8$ are 108 ms and 71 ms.

50 μs CT. Fig. 14 shows the mean end-to-end packet delays for both ST and BE packets. We observe from Fig. 14(a) that for the low injection rate $\pi = 2$ ST packets/CT, the ST packet delays were minuscule for both ATS and TAS. We observe from Fig. 14(b) that for the higher $\pi = 4$ ST packets/CT injection rate, TAS continues to ensure very low ST packet delays across the entire load range. In contrast, we observe that the ATS ST packet delay increases exponentially with the load; ATS provides mean ST packet delays below 1 ms only for loads below $\rho_L = 1$. With ATS, the resource allocation is fixed and the ATS shapers and urgent queues become saturated when the ST injection rate is $\pi = 4$. We further observe from Fig. 14(c) that for the high ST injection rate $\pi = 8$, ATS provides sub-millisecond mean packet delays up to a load around $\rho_L = 0.8$; whereas, TAS gives delays above 50 ms consistently for all load levels. Intuitively, the inherently asynchronous ATS struggles with moderately high to high ($\pi = 4$ and 8) periodic (synchronous) ST packet traffic as the asynchronous ATS prioritization mechanisms do not work in lock-step with the traffic sources. Thus, the ATS delay performance degrades gradually as the ST and BE traffic loads increase. On the other hand, the inherently synchronous TAS can either consistently provide sub-millisecond ST packet delays ($\pi = 2$ and 4), or completely fails ($\pi = 8$), even for low loads of competing BE traffic. This abrupt failure of TAS is due to the prescribed gating ratio that is synchronized to the underlying cycle time; the ST traffic either fits into the ST portion of the cycle (or not) and thus either meets real-time requirements (or not). In contrast, the asynchronous ATS degrades gradually as the competing BE traffic load increases.

Fig. 15 shows the maximum packet delay experienced within the network. The results generally mirror the mean delay results in Fig. 14 in that for $\pi = 2$, all schemes give minuscule maximum packet delays well below one millisecond. For $\pi = 4$, the maximum ATS delay increases with the BE traffic load, reaching 10 ms for a BE traffic load around $\rho_L = 0.9$, while TAS continues for provide minuscule maximum delays. For $\pi = 8$, TAS gives very high delays on the order of 100 ms, while ATS gives around

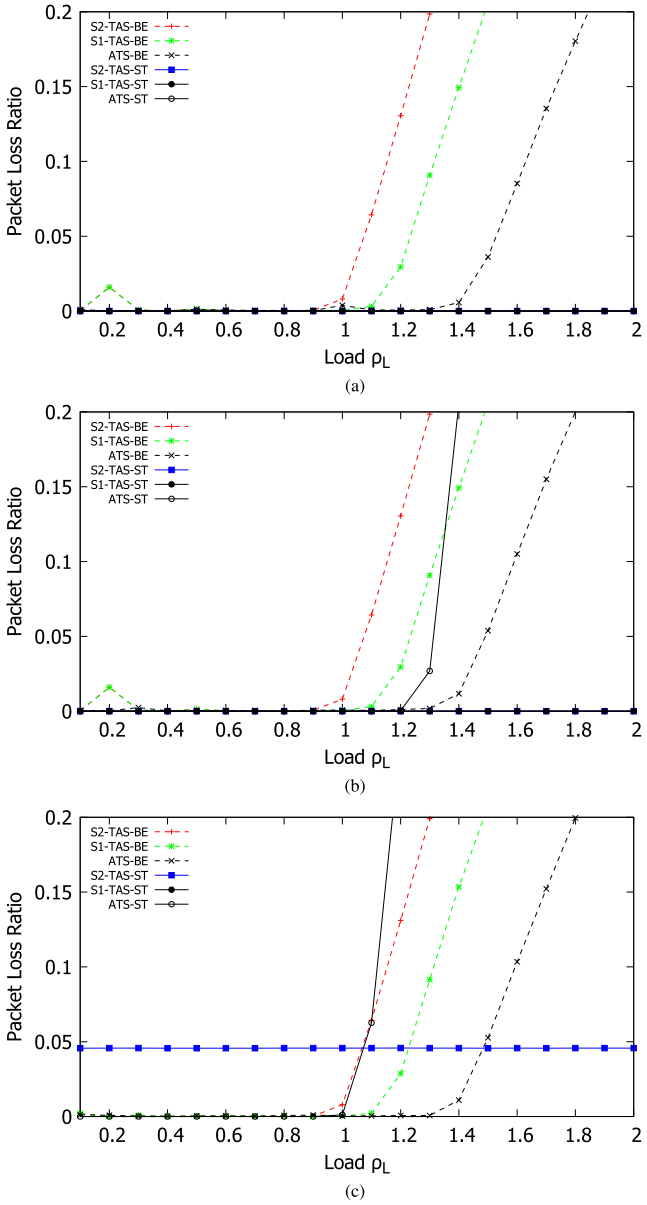


FIGURE 16. Packet loss ratio for ATS compared with standard TAS. (a) $\pi = 2$ ST packets/CT, (b) $\pi = 4$ ST packets/CT, and (b) $\pi = 8$ ST packets/CT.

10 ms maximum delay for a load of $\rho = 0.6$. Essentially, these results are again due to the ATS state machine becoming gradually overwhelmed as the load increases, whereas TAS either fits the ST traffic into the ST gate window or not.

Fig. 16 shows the packet loss ratios for both BE and ST traffic with ATS compared to standard TAS. We observe that similar to the mean delay behaviors in Fig. 14, (*i*) the ATS and TAS ST packet loss ratios are zero for the low ST traffic rate $\pi = 2$, (*ii*) for the moderate ST traffic rate $\pi = 4$, ATS starts to drop ST packets at a moderately high BE traffic load while TAS still achieves consistently zero losses, and (*iii*) for the high $\pi = 8$ ST traffic rate, S1 TAS gives 0.37 (37%) loss while S2 TAS gives a loss ratio close to 0.05 consistently for all BE load levels; in contrast, ATS gives zero ST losses for low BE loads and then increasing ST losses for moderately

high BE loads. We also observe from Fig. 16 that ATS gives lower BE packet losses than TAS for all considered scenarios.

The explanation for these loss ratio behaviors is similar to the explanation of the mean delay behaviors. Essentially, the asynchronously operating ATS gradually degrades with increasing BE traffic load when the synchronous ST traffic is moderately high to high. In other words, each ATS switch operates individually in complete isolation from the other switches and the sources In contrast, all the TAS switches are synchronized to the common cycle time that is the underlying time basis for the periodic ST packet transmissions by the sources.

3) SUMMARY OF PACKET DELAY VARIATION RESULTS

Compared to the packet delay variations of TAS (see Section IV-B.3), ATS gave generally higher packet delay variations. With ATS, sporadic ST packets experienced delay variations (standard deviations of packet delays) ranging from 0.1 ms for low loads to over 0.6 ms for moderate to high loads. Periodic ST packets experienced delay variations ranging from 0.4 ms to 50 ms for ST packet injection rates of $\pi = 2$ and 4 with moderate to high BE loads.

V. CONCLUSIONS AND FUTURE WORK

This article has examined the Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS) mechanisms of the 802.1 Time Sensitive Networking (TSN) standard. In order to address the TAS shortcomings, we have proposed two novel mechanisms, namely Adaptive Bandwidth Sharing (ABS) and an Adaptive Slot Window (ASW) mechanisms. We have evaluated the TSN standard mechanisms and the novel mechanisms through extensive simulations with both sporadic Poisson traffic and periodic traffic. We found that standard TAS generally achieves ultra-short latencies if the gating ratio for high-priority scheduled traffic (ST) is sufficiently large to accommodate the high-priority traffic volume. We also observed that the introduced ABS mechanism enhances the quality of service provided to low-priority best effort (BE) traffic, while maintaining the ultra-short latencies for high-priority traffic. The introduced ASW mechanism dynamically adjusts the gating ratio for high-priority traffic so as to ensure ultra-low latencies for high-priority traffic, even for fluctuating background traffic loads of low-priority traffic.

We also found that the asynchronous ATS performs generally well compared to TAS for sporadic (asynchronous traffic). However, for periodic ST traffic with moderately high rates, ATS gives increasing ST packet delays for increasing loads of competing BE traffic. In contrast, TAS with time synchronization to the underlying period of the ST traffic sources either provides very short (order of milliseconds or less) delays irrespective of the BE traffic load, or has consistently high delays (on the order of 100 ms) when the configured gate ratio is too small for the ST traffic.

There are numerous opportunities for future TSN research that we proceed to outline. Combining the Adaptive Bandwidth Sharing (ABS) and the Adaptive Slotted Window (ASW) TAS enhancements, i.e., Adaptive TAS, with frame/packet preemption will likely achieve further delay reductions. The delay reductions will be particularly relevant for priority inversion scenarios, i.e., to reduce the delay of high-priority ST traffic that would be blocked due to slot allocations to BE traffic. Additionally, modifying the ST gating ratio through updates ranging from course-grained increments/decrements to fine-grained values relative to network conditions may achieve further performance enhancements. The present study examined static networking scenarios in that a given evaluation run considered a fixed prescribed traffic load. Future research may consider transient TSN network scenarios with varying traffic loads, e.g., scenarios that connect new and remove old periodic and sporadic ST sources during runtime.

We believe that another important future research direction will be the integration of the TSN network control with the emerging universal SDN control of communication networks [60]–[63]. It will be critical to define standardized interfaces that facilitate SDN control down to the TSN TAS gating level. The gate operation should still be tied to the time synchronization (which can run independently of the SDN control). However, the specific actions and quantities of slot durations that follow upon time synchronization points should be under SDN control. For instance, SDN control should be able to obtain the utilization level of the various slots, and then be able to adjust gating ratios.

REFERENCES

[1] S. H. Hong, J. Sun, M. Yu, X. Zhang, and A. Xu, "Collaborative coexistence management scheme for industrial wireless sensor networks," *IEEE Access*, vol. 7, pp. 1617–1626, 2019.

[2] A. Karimi, K. I. Pedersen, N. H. Mahmood, J. Steiner, and P. Mogensen, "5G centralized multi-cell scheduling for URLLC: Algorithms and system-level performance," *IEEE Access*, vol. 6, pp. 72253–72262, 2018.

[3] S. Yang, P. Wieder, M. Aziz, R. Yahyapour, X. Fu, and X. Chen, "Latency-sensitive data allocation and workload consolidation for cloud storage," *IEEE Access*, vol. 6, pp. 76098–76110, 2018.

[4] K. Lee, J. Kim, Y. Park, H. Wang, and D. Hong, "Latency of cellular-based V2X: Perspectives on TTI-proportional latency and TTI-independent latency," *IEEE Access*, vol. 5, pp. 15800–15809, 2017.

[5] M. A. Lema *et al.*, "Business case and technology analysis for 5G low latency applications," *IEEE Access*, vol. 5, pp. 5917–5935, 2017.

[6] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, vol. 6, pp. 12825–12837, 2018.

[7] G. Pocovi, K. I. Pedersen, and P. Mogensen, "Joint link adaptation and scheduling for 5G ultra-reliable low-latency communications," *IEEE Access*, vol. 6, pp. 28912–28922, 2018.

[8] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55765–55779, 2018.

[9] G. M. S. Rahman, M. Peng, K. Zhang, and S. Chen, "Radio resource allocation for achieving ultra-low latency in fog radio access networks," *IEEE Access*, vol. 6, pp. 17442–17454, 2018.

[10] M. Waqar, A. Kim, and P. K. Cho, "A transport scheme for reducing delays and jitter in Ethernet-based 5G fronthaul networks," *IEEE Access*, vol. 6, pp. 46110–46121, 2018.

[11] M. A. Al-Maqri, M. A. Alrshah, and M. Othman, "Review on QoS provisioning approaches for supporting video traffic in IEEE802.11e: Challenges and issues," *IEEE Access*, vol. 6, pp. 55202–55219, 2018.

[12] A. Kawabata, B. C. Chatterjee, S. Ba, and E. Oki, "A real-time delay-sensitive communication approach based on distributed processing," *IEEE Access*, vol. 5, pp. 20235–20248, 2017.

[13] A. Nasrallah *et al.*, "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 88–145, 1st Quart., 2019.
[14] E. Raza *et al.*, "Dynamic priority based reliable real-time communications for infrastructure-less networks," *IEEE Access*, vol. 6, pp. 67338–67359, 2018.
[15] X. Jiang *et al.*, "Low-latency networking: Where latency lurks and how to tame it," *Proc. IEEE*, vol. 107, no. 2, pp. 280–306, Feb. 2019.
[16] F. Norman. (Jul. 2017). *Time-Sensitive and Deterministic Networking Whitepaper*. Accessed: Feb. 19, 2019. [Online]. Available: https://mentor.ieee.org/802.24/dcn/17/24-17-0020-00-sgtg-contribution-time-sensitive-and-deterministic-networking-whitepaper.pdf,
[17] R. Makowitz and C. Temple, "FlexRay–a communication network for automotive control systems," in *Proc. IEEE Int. Workshop Factory Commun. Syst.*, 2006, pp. 207–212.
[18] *IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks—Amendment 25: Enhancements for Scheduled Traffic*, Standard IEEE 802.1Qbv-2015, Mar. 2016, pp. 1–57.
[19] J. Specht, *IEEE Draft Standard for Local and metropolitan area networks–Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Asynchronous Traffic Shaping*, Standard IEEE P802.1Qcr/D0.5, Jun. 2018.
[20] D. Shrestha, Z. Pang, and D. Dzung, "Precise clock synchronization in high performance wireless communication for time sensitive networking," *IEEE Access*, vol. 6, pp. 8944–8953, Feb. 2018.
[21] H. Kopetz, A. Ademaj, P. Grillinger, and K. Steinhammer, "The time-triggered Ethernet (TTE) design," in *Proc. IEEE Int. Symp. Object-Oriented Real-Time Distr. Comput. (ISORC)*, May 2005, pp. 22–33.
[22] *IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks–Amendment 28: Per-Stream Filtering and Policing*, Standard IEEE 802.1Qci-2017, Sep. 2017, pp. 1–65.
[23] *IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks–Amendment 29: Cyclic Queuing and Forwarding*, Standard IEEE 802.1Qch-2017, Jun. 2017, pp. 1–30.
[24] P. Pedreiras, L. Almeida, and P. Gai, "The FTT-Ethernet protocol: Merging flexibility, timeliness and efficiency," in *Proc. IEEE Eur. Conf. Real-Time Syst.*, Jun. 2002, pp. 134–142.
[25] P. Pedreiras, P. Gai, L. Almeida, and G. C. Buttazzo, "FTT-Ethernet: A flexible real-time communication protocol that supports dynamic QoS management on Ethernet-based systems," *IEEE Trans Ind. Informat.*, vol. 1, no. 3, pp. 162–172, Aug. 2005.
[26] P. Meyer, T. Steinbach, F. Korf, and T. C. Schmidt, "Extending IEEE 802.1 AVB with time-triggered scheduling: A simulation study of the coexistence of synchronous and asynchronous traffic," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2013, pp. 47–54.
[27] S. Thangamuthu, N. Concer, P. J. L. Cuijpers, and J. J. Lukkien, "Analysis of Ethernet-switch traffic shapers for in-vehicle networking applications," in *Proc. IEEE Design, Autom. Test Eur. Conf. Exhibit.*, Mar. 2015, pp. 55–60.
[28] D. Thiele, R. Ernst, and J. Diemer, "Formal worst-case timing analysis of Ethernet TSN's time-aware and peristaltic shapers," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2015, pp. 251–258.
[29] C. Park, J. Lee, T. Tan, and S. Park, "Simulation of scheduled traffic for the IEEE 802.1 time sensitive networking," in *Information Science and Applications* (Lecture Notes in Electrical Engineering), vol. 376. Singapore: Springer, 2016, pp. 75–83.
[30] M. H. Farzaneh and A. Knoll, "Time-sensitive networking (TSN): An experimental setup," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Nov. 2017, pp. 23–26.
[31] D. Maxim and Y.-Q. Song, "Delay analysis of AVB traffic in time-sensitive networks (TSN)," in *Proc. ACM Int. Conf. Real-Time Netw. Syst. (RTNS)*, Oct. 2017, pp. 18–27.
[32] S. Nsaibi, L. Leurs, and H. D. Schotten, "Formal and simulation-based timing analysis of industrial-Ethernet sercos III over TSN," in *Proc. IEEE/ACM Int. Symp. Distrib. Simulation Real Time Appl. (DS-RT)*, Oct. 2017, pp. 1–8.
[33] L. Zhao, P. Pop, and S. S. Craciunas, "Worst-case latency analysis for IEEE 802.1Qbv time sensitive networks using network calculus," *IEEE Access*, vol. 6, pp. 41803–41815, 2018.
[34] F. Dürr and N. G. Nayak, "No-wait packet scheduling for IEEE time-sensitive networks (TSN)," in *Proc. ACM Int. Conf. Real-Time Netw. Syst.*, 2016, pp. 203–212.
[35] S. S. Craciunas, R. S. Oliver, M. Chmelík, and W. Steiner, "Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks," in *Proc. ACM Int. Conf. Real-Time Netw. Sys.*, 2016, pp. 183–192.
[36] S. S. Craciunas, R. S. Oliver, and W. Steiner. (2017). "Formal scheduling constraints for time-sensitive networks." [Online]. Available: https://arxiv.org/abs/1712.02246
[37] M. Lander, P. Raagaard, M. G. Pop, M. Gutiérrez, and W. Steiner, "Run-time reconfiguration of time-sensitive networking (TSN) schedules for fog computing," in *Proc. IEEE Fog World Congress*, Oct. 2017, pp. 1–6.
[38] V. Gavriluţ, L. Zhao, M. L. Raagaard, and P. Pop, "AVB-aware routing and scheduling of time-triggered traffic for TSN," *IEEE Access*, vol. 6, pp. 75229–75243, 2018.
[39] N. G. Nayak, F. Dürr, and K. Rothermel, "Routing algorithms for IEEE802.1Qbv networks," *ACM SIGBED Rev.*, vol. 15, no. 3, pp. 13–18, Jun. 2018.
[40] N. G. Nayak, F. Dürr, and K. Rothermel, "Incremental flow scheduling and routing in time-sensitive software-defined networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 2066–2075, May 2017.
[41] T. Wan, B. McCormick, Y. Wang, and P. Ashwood-Smith, "ZeroJitter: An SDN based scheduling for CPRI over Ethernet," in *Proc. IEEE GLOBECOM*, Dec. 2016, pp. 1–7.
[42] D. Hisano *et al.*, "Gate-shrunk time aware shaper: Low-latency converged network for 5G fronthaul and M2M services," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.
[43] T. Wan and P. Ashwood-Smith, "A performance study of CPRI over Ethernet with IEEE 802.1Qbu and 802.1Qbv enhancements," in *Proc. IEEE Globecom*, Dec. 2015, pp. 1–6.
[44] M. K. Al-Hares, P. Assimakopoulos, D. Muench, and N. J. Gomes, "Scheduling in an Ethernet fronthaul network," in *Proc. IEEE Eur. Conf. Netw. Commun.(EuCNC)*, Jun. 2017, pp. 1–5.
[45] M. K. Al-Hares, P. Assimakopoulos, D. Muench, and N. J. Gomes, "Modeling time aware shaping in an Ethernet fronthaul," in *Proc. IEEE GLOBECOM*, Dec. 2017, pp. 1–6.
[46] M. K. Al-Hares, P. Assimakopoulos, D. Muench, and N. J. Gomes, "Traditional queuing regimes and time-aware shaping performance comparison in an Ethernet fronthaul network," in *Proc. IEEE Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2017, pp. 1–4.
[47] P. Assimakopulos, G. S. Birring, M. K. Al-Hares, and N. J. Gomes, "Ethernet-based fronthauling for cloud-radio access networks," in *Proc. IEEE Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2017, pp. 1–4.
[48] L. L. Bello, "Novel trends in automotive networks: A perspective on Ethernet and the IEEE Audio Video Bridging," in *Proc. IEEE Emerg. Technol. Factory Automat. (ETFA)*, Sep. 2014, pp. 1–8.
[49] Z. Zhou, Y. Yan, M. Berger, and S. Ruepp, "Analysis and modeling of asynchronous traffic shaping in time sensitive networks," in *Proc. IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, Jun. 2018, pp. 1–4.
[50] J.-Y. L. Boudec. (2018). "A theory of traffic regulators for deterministic networks with application to interleaved regulators." [Online]. Available: https://arxiv.org/abs/1801.08477
[51] E. Mohammadpour, E. Stai, M. Mohiuddin, and J.-Y. Le Boudec. (2018). "End-to-end latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping." [Online]. Available: https://arxiv.org/abs/1804.10608
[52] *IEEE Standard for Local and Metropolitan Area Networks–Frame Replication and Elimination for Reliability* Standard IEEE Std 802.1CB-2017, Oct. 2017, pp. 1–102.
[53] S. S. Craciunas and R. S. Oliver, "An overview of scheduling mechanisms for time-sensitive networks," Real-time Summer School, LEcoledEte Temps Reel (ETR), Paris, France, Tech. Rep., 2017.
[54] J. Specht and S. Samii, "Urgency-based scheduler for time-sensitive switched Ethernet networks," in *Proc. IEEE Euromicro Conf. Real-Time Syst.*, Jul. 2016, pp. 75–85.
[55] F. Chen, F.-J. Goetz, M. Kiessling, and J. Schmitt. (Dec. 2018). *Aggregation of micro-streams into one common stream*. Accessed: Dec. 20, 2018. [Online]. Available: http://ieee802.org/1/files/public/docs2018/dd-chen-flow-aggregation-1218-v02.pdf
[56] J. W. Guck, M. Reisslein, and W. Kellerer, "Function split between delay-constrained routing and resource allocation for centrally managed QoS in industrial networks," *IEEE Trans. Ind. Informat.*, vol. 12, no. 6, pp. 2050–2061, Dec. 2016.
[57] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proc. ICST Int. Conf. Simulation Tools Techn. Commun., Netw. Syst. Workshops*, 2008, pp. 1–10.

[58] R. Belliardi *et al.* (Sep. 2018). *Use Cases IEC/IEEE 60802, Version 1.3*. Accessed: Feb. 19, 2019. [Online]. Available: http://www.ieee802.org/1/files/public/docs2018/60802-industrial-use-cases-0918-v13.pdf
[59] *IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks—Amendment 26: Frame Preemption*, Standard IEEE Std 802.1Qbu-2016, Aug. 2016, pp. 1–52.
[60] I. Benacer, F.-R. Boyer, and Y. Savaria, "A high-speed, scalable, and programmable traffic manager architecture for flow-based networking," *IEEE Access*, vol. 7, pp. 2231–2243, 2019.
[61] J. H. Cox *et al.*, "Advancing software-defined networks: A survey," *IEEE Access*, vol. 5, pp. 25487–25526, 2017.
[62] W. Kellerer, P. Kalmbach, A. Blenk, A. Basta, M. Reisslein, and S. Schmid, "Adaptable and data-driven softwarized networks: Review, opportunities, and challenges," *Proc. IEEE*, vol. 107, no. 4, pp. 711–731, Apr. 2019.
[63] Q.-Y. Zhang, X.-W. Wang, M. Huang, K.-Q. Li, and S. K. Das, "Software defined networking meets information centric networking: A survey," *IEEE Access*, vol. 6, pp. 39547–39563, 2018.

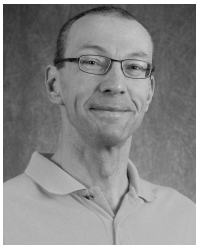
AHMED NASRALLAH received the B.Sc. degree in electrical and computer engineering from the University of Dayton, Dayton, OH, USA, and the M.S. degree in computer engineering from Arizona State University, Tempe, where he is currently pursuing the Ph.D. degree. He is currently a Researcher funded by Kuwait University. His research interests include communication and multimedia networking.

AKHILESH S. THYAGATURU received the Ph.D. degree in electrical engineering from Arizona State University (ASU), Tempe, in 2017. He was with Qualcomm Technologies, Inc., San Diego, CA, USA, as an Engineer, from 2013 to 2015. He is currently an Engineer with Intel Corporation, Chandler, AZ, USA, and an Adjunct Faculty with the School of Electrical, Computer, and Energy Engineering, ASU. He serves as a Reviewer for various journals, including the IEEE COMMUNICATIONS SURVEYS & TUTORIALS, the IEEE TRANSACTIONS OF NETWORK AND SERVICE MANAGEMENT, and *Optical Fiber Technology*.

ZIYAD ALHARBI received the B.Sc. degree in electrical engineering from the King Fahd University of Petroleum and Minerals, Saudi Arabia, and the M.S. degree in electrical engineering from Arizona State University, Tempe, where he is currently pursuing the Ph.D. degree. He is currently a Researcher with King Abdulaziz City for Science and Technology, Riyadh, Saudi Arabia. He serves as a Reviewer for various journals, including the IEEE COMMUNICATIONS SURVEYS & TUTORIALS, *Computer Networks*, and *Optical Switching and Networking*.

CUIXIANG WANG received the B.Sc. degree in computer science and technology from Qufu Normal University, Qufu, China, in 2007, and the M.S. degree in computer software and theory from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2010. She is currently a Lecturer with the School of Information Engineering, Yancheng Institute of Technology, Yancheng, China. She is also a Visiting Scholar with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe. Her research interests include wireless network routing and network simulation.

XING SHAO received the Ph.D. degree in information network from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2013. He is currently an Associate Professor with the School of Information Engineering, Yancheng Institute of Technology, Yancheng, China. He is also a Visiting Scholar with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe. His research interests include wireless multihop networking, routing, and network simulation.



MARTIN REISSLEIN (S'96–M'98–SM'03–F'14) received the Ph.D. degree in systems engineering from the University of Pennsylvania, in 1998. He is currently a Professor with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON EDUCATION, IEEE ACCESS, and *Computer Networks*. He is an Associate Editor-in-Chief for the IEEE COMMUNICATIONS SURVEYS & TUTORIALS, a Co-Editor-in-Chief of *Optical Switching and Networking*, and chairs the Steering Committee of the IEEE TRANSACTIONS ON MULTIMEDIA.

HESHAM ELBAKOURY received the M.Sc. degree from Waterloo University, ON, Canada. He was a Chief Systems Architect with the Hitachi-CTA EPON Access Systems Division, and a Chief Systems Architect with Nortel and Bell-Northern Research, where he led the architecture, design, and development of several very successful Switching/Routing, Security and Carrier Ethernet products. In Nortel/BNR, he initiated and led the Autonomic Network research project in the Enterprise Division, and the Software Design and Code Reuse Project in the Data Networking Division. He is a 35 year veteran in the telecommunications and data networking industry with an extensive background and expertise in the architecture, design, and development of Distributed Systems and Broadband Access, Enterprise and Telco Communications Systems. He is currently a Principal Architect with Futurewei Technologies, Inc., focusing on advanced technology research and standards in the Network Research Lab. He has been active in different standard groups, including IEEE 802, IEEE 1904, IETF, ONF, OIF, MEF, the SCTE Energy 2020 program, and CableLabs where he has been heavily involved in IEEE 802.3/802.1, DPoE/DPoG, DOCSIS 3.1, Full-Duplex DOCSIS 3.1, SDN/NFV, Distributed CCAP Architectures, and Business Services projects.