Survey, Design and Evaluation of TGT-HC: A Time-Aware Shaper MAC for Wireless TSN

Raymond J. Jayabal, *Member, IEEE*, David Tung Chong Wong, *Senior Member, IEEE*, Lee Kee Goh, Xiaojuan Zhang, *Member, IEEE*, Chin Ming Pang, and Sumei Sun, *Fellow, IEEE*

***Abstract*—Ultra-Reliable Low-Latency Communication (URLLC) and Time-Sensitive Networking (TSN) are essential for enhancing 5G and Wi-Fi 6/7 to support real-time industrial automation. However, our survey shows that existing Medium Access Control (MAC) schemes still face unresolved latency issues. This paper introduces the Transmission Gating Time Hyperchannel (TGT-HC), a novel contention-free Carrier-Sense Multiple Access (CSMA) scheme driven by a per-flow Time-Aware Shaper (TAS) scheduler. Our analytical results, simulations, and prototyping with the Universal Software Radio Peripheral (USRP) demonstrate that TGT-HC achieves latency performance comparable to a First-Come-First-Served (FCFS) single server for real-time cyclic traffic, even under high frame error rates (FERs). Given its promising performance, we advocate for reconsidering contention-free CSMA as a viable MAC scheme in next-generation URLLC/TSN.**

***Index Terms*—Ultra-reliable low-latency communication (URLLC), time-sensitive networking (TSN), medium access control (MAC), contention-free CSMA, transmission gating time hyperchannel (TGT-HC), time-aware shaper (TAS), real-time cyclic traffic, industrial wireless networks.**

I. Introduction

In REAL-TIME industrial control applications, achieving ultra-reliable, low-latency wireless communication remains a significant challenge. Wireless systems like 5G and Wi-Fi, while promising, often fall short of wired reliability due to higher latency and susceptibility to interference. To address these issues, substantial research has been directed toward developing new Time-Sensitive Networking (TSN) [1] and Ultra-Reliable Low-Latency Communications (URLLC) [2] schemes for existing Medium Access Control (MAC) protocols [3]. While existing works focus on Time Division Multiple Access (TDMA) [2], [4] and polling MAC schemes [1], [5], [6], we diverge by introducing a novel approach based on contention-free Carrier-Sense Multiple Access (CSMA).

Received 16 January 2024; revised 12 November 2024; accepted 16 January 2025. This work was supported by the A*STAR Industrial Internet of Things Research Program under the RIE2020 IAF-PP under Grant A1788a0023. Recommended for acceptance by N. Zhang. *(Corresponding author: Raymond J. Jayabal.)*

Raymond J. Jayabal, Lee Kee Goh, Xiaojuan Zhang, Chin Ming Pang, and Sumei Sun are with the Communications & Networks Department, Institute for Infocomm Research, A*STAR, Singapore 138632 (e-mail: jraymond@i2r.a-star.edu.sg; lkgoh@i2r.a-star.edu.sg; xzhang@i2r.a-star.edu.sg; pangcm@i2r.a-star.edu.sg; sunsm@i2r.a-star.edu.sg).

David Tung Chong Wong was with the Communications & Networks Department, Institute for Infocomm Research, A*STAR, Singapore 138632. He is now an Independent Researcher, Singapore 534207 (e-mail: davidwongtungchong@gmail.com).

Digital Object Identifier 10.1109/TMC.2025.3535413

As our survey reveals, contention-free CSMA offers shorter channel access delays than TDMA due to its deterministic access arbitration using small sensing slots. It also avoids the polling overhead in polling MAC schemes. Specifically, we focus on the Hyperchannel [7] scheme, a contention-free CSMA protocol which assigns sensing slots based on fixed priorities. By enhancing it with a Time-Aware Shaper (TAS) [8] scheduler to assign sensing slots based on transmission gating windows (TGWs), we achieved analytical, simulation, and prototyping results close to the First-Come-First-Served (FCFS) single server, even at a 20% frame error rate (FER).

The main contribution of this paper is the design and evaluation of our Transmission Gating Time Hyperchannel (TGT-HC) scheme. To our knowledge, no other wireless TSN MAC scheme comes close to matching the ideal FCFS delay performance as ours. This was achieved by restricting CSMA participation to only the nodes with active TGWs at any given moment, and employing specialized slotting sequences that allow flows in overlapping TGWs to transmit immediately after one another without arbitration. Additionally, we include a mini-survey that identifies key challenges and solutions for integrating the TAS with TDMA, polling-MAC and contention-based CSMA schemes. We also compare the delay performances of various contention-free CSMA protocols in handling cyclic traffic. Finally, the validation of our results with Universal Software Radio Pheripheral (USRP) hardware represents perhaps the strongest contribution of our paper.

The rest of the paper is structured as follows: Section II reviews the TAS scheduler, presents the mini-survey, and describes the Hyperchannel scheme that our work enhances. Section III details the design of TGT-HC. Sections IV and V provide analytical and simulation results. Section VI presents the USRP-based prototyping results and compares them with results from TSN-enabled Wi-Fi 6/7. Finally, Section VII summarizes the key achievements and outlines future plans for this protocol. This paper expands upon our earlier work [9] by adding a mini-survey in Section II, a detailed explanation of the slotting sequences in Section III, an analytical evaluation for non-zero FER in Section IV, and the prototyping results alongside their comparisons against Wi-Fi 6/7 in Section VI.

II. Background Survey

A. TAS Scheduler & Related Works

In the IEEE TSN suite [3], the TAS scheduler [8] is the component that ensures low-latency and reliable delivery of real-time scheduled traffic. It is based on the Time-Triggered

1536-1233 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

IEEE Proof

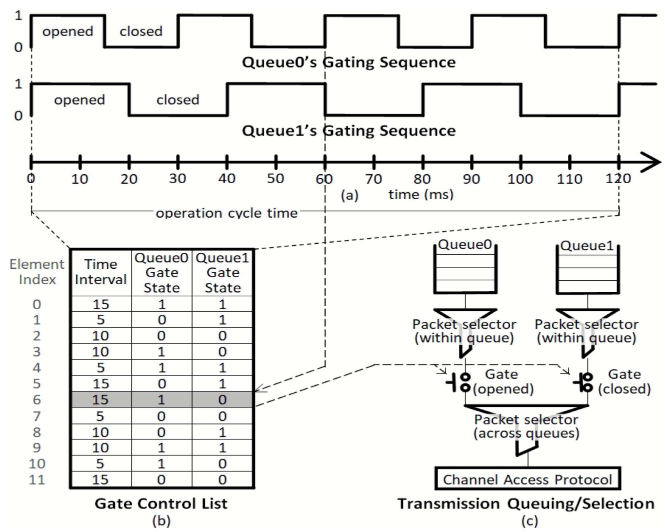

Fig. 1. TAS scheduler scheme.

Protocol (TTP) [10] where packet arrival times are assumed known a priori. As a result, TDMA slots on each link in packet's route can be scheduled closely together, reducing its delay through the network. But while the TTP is a channel access protocol that schedules packet transmissions, the TAS operates above these protocols by managing gates that control packets from which queues are chosen for transmission.

To show how this works, consider two queues handling cyclical traffic flows: packets arrive every 30 ms at Queue0 and every 40 ms at Queue1 (Fig. 1). Accordingly, the TAS opens and closes gates in sync with the packet arrivals (Fig. 1(a)). Although in this example the gating durations are half the arrival intervals, in reality they depend on factors like physical transmission durations, inter-frame gaps, acknowledgments, retries, maximum latency, and packet lifetime.

The gating sequences are recorded in a Gate Control List (GCL). Each entry in this list specifies the gating state and duration for each queue in chronological order. The timespan covered by this list is until the combined gating sequence of the queues repeats, which is 120 ms in this case (Fig. 1(b)). This list is used cyclically to permit or deny packet selection from the queues for transmission (Fig. 1(c)).

To ensure the accuracy and timeliness of real-time data streams across the entire network, each routing node must have its GCL configured such that, as soon as a real-time packet is completely received from the previous node, the corresponding gating window opens in the current node to allow the packet to be immediately transmitted to the next node. By scheduling the gate openings and closings to align with the traffic flows as they traverse the nodes, queuing delays are minimized, helping the packets meet their strict deadlines.

Through various scheduling algorithms such as Fixed-Priority Scheduling (FPS), Earliest Deadline First (EDF), and Rate Monotonic Scheduling (RMS), the GCLs can also be configured to transmit packets with tighter deadlines earlier. For instance, consider a scenario where Queue0's packets have a deadline of 10 ms and arrive at the start of the GCL sequence, while Queue1's packets have a deadline of 5 ms and arrive at the 1 ms mark. Assuming the packet transmission durations are longer than 1 ms, then based on the EDF algorithm, the first gating window for the GCL will be configured to open at the 1 ms mark for Queue1's packets even though Queue0's packets arrive earlier, as Queue1's packets have a shorter deadline.

To date, the TAS scheduler has been implemented on several network simulation platforms [11]. Latency analyses have been conducted comparing it to other schedulers such as TTE [12] and Asynchronous Traffic Shaper (ATS) [13]. Its performance in regulating real-time traffic for 5G digital Common Public Radio Interface (CPRI) signalling [14] has also been evaluated. Additionally, heuristic methods to generate GCLs with minimal latency and jitter have been proposed [15]. Apart from heuristics, deep reinforcement learning techniques such as Deep Q-Learning (DQN) have also been shown to enhance flow prioritization and scheduling [16], [17], suggesting that similar approaches can also be applied to GCL configurations.

But the area most relevant to our research is the integration of TAS with MAC protocols. Specifically, we focused on parametric optimizations, special modes or enhanced schemes that could reduce the delays and overprovisioning of the gating windows, especially over wireless protocols where retransmissions are frequent. In this context, some relevant work has been conducted within the 802.11be standardization group: In [18], simulations of the TAS over the 802.11ax Trigger-Based (TB) Multi-User (MU) protocol showed better latency Cumulative Density Functions (CDFs) compared to Enhanced Distributed Channel Access (EDCA). In [19], [20], pausing the EDCA contention during the gating windows for real-time traffic and limiting its TXOP window were considered. In [5], reducing the 802.11ax trigger-based access overhead was identified as a long-term goal. One approach was to send semi-persistent schedules (SPS) in announcement frames only when uplink schedule changes occur, rather than repeatedly sending instantaneous uplink (UL) allocations on trigger frames before each UL transmission [6]. Outside the 802.11 standardization, a prototype TDMA protocol with TAS-like scheduling on Software Defined Radios (SDRs) was presented in [4]. While similar to our work, it is based on TDMA whereas our design is based on contention-free CSMA.

B. *TDMA Issues & Related Works*

TDMA is commonly used in various industrial [21], [22] and experimental [4], [23], [24] wireless systems. In this scheme, nodes transmit in the timeslots within a repeating time interval known as the superframe. Although not strictly TDMA, LTE/5G [2] employs a similar slot and superframe scheme. Consequently it also faces TDMA's two main issues when supporting TAS scheduling: The first issue arises when a TAS gating window starts just after a slot boundary, making the next available slot to be at least a whole slot away. To avoid this slot boundary delay, the windows' intervals and starting offsets can be restricted to align to integral multiples of the slot [4]. However this would also restrict the range of flow intervals and starting offsets that can be supported. Another approach is to shorten the timeslots by limiting the maximum number of transmission symbols within and/or reducing the symbol duration, as done for LTE/5G [25]. But this would require large packets to be split over multiple slots, adding overheads such as PHY preambles and MAC headers per segment. Shorter symbol durations also

IEEE Proof

weaken multipath resilience, making it less viable for factory scenarios. To avoid these setbacks, slot timings that match packet arrival times and transmission durations were proposed in [26], [27].

The other key issue with TDMA is that control messages are required to change slot allocations. These control messages not only introduce messaging delay but can also lead to slots being unused during idle periods [28] or when slots are reserved for retransmission that do not occur. In LTE/5G systems, extra slots for retransmissions are assigned via on-the-fly grant messages [29]. Initially, the round-trip time (RTT) between retransmission control messages was reduced by shortening the slot duration [30]. Later, it was more significantly reduced with the mini-slot scheme [31], which allows uplink and downlink bursts within the same slot. In [21], allocating slots for retransmissions is avoided by provisioning them in the initial flow allocation itself. In [22], multiple retransmissions are scheduled in the same slot over multiple available radio channels. In both these cases, the extra slots are wasted if retransmission is not needed. To address this, [32] suggested pooling SPS allocations for retransmissions among multiple nodes, while [33] proposed statistical sharing of slots among nodes without significantly increasing collision rates.

C. *Polling MAC Issues & Related Works*

The polling MAC scheme is utilized in several Wi-Fi standards, including the TSN-enabled IEEE 802.11be [1], [34], [35]. In this scheme, a node transmits a data frame only after receiving a poll frame from a central node. If the node has no data to send, it sends a null frame. The central node then polls the next node in turn. In some variants, the polled node can send or resend multiple data frames per poll, and concludes its turn by keeping the channel idle for a specified duration. This scheme overcomes the two main issues of TDMA: With poll frames properly timed in advance, packets can be transmitted just as the gating windows begin. Whereas in TDMA, this can only happen if the gating windows and slots are synchronized. Additionally, the polled node can terminate the granted airtime, allowing it to release the channel for idle periods or extend it for retransmissions without additional control messaging as required in TDMA.

This scheme faces two issues, with the first being the poll frame overhead. In the TAS context, shortening the poll frame [6] decreases the airtime needed for its transmission before each gating window. This reduction allows the windows for consecutive flows to be packed more closely together. As the poll frames are also prone to reception errors, reducing their length is also crucial, especially when they have to be retransmitted. For cyclic flows, polling can be significantly reduced: In [36] for example, the "more data" bit in the 802.11 header allows a node to transmit its next packet without needing another poll. In [1], schedules were added to the trigger frame granting the node to transmit in subsequent cycles. A single poll frame can also address multiple nodes. For example, in IEEE 802.11ax/be, the trigger frame assigns separate Orthogonal Frequency Division Multiple Access (OFDMA) resource units (RUs) to multiple nodes. Whereas in [37], the poll frame assigns TDMA slots to multiple nodes. In [38], short OFDMA subcarrier signals were utilized in place of the poll frame to further reduce overhead. However, using a multi-node poll frame increases the likelihood that at least one node receives it erroneously. For example, with a 5% FER, a 4-node poll frame has an 18% chance of being discarded by at least one node due to errors, compared to just 5% for a 1-node poll frame. As a result, more nodes will miss transmitting within their gating windows, negatively impacting the delay bound reliability. Furthermore, using RUs or TDMA slots also brings about the TDMA issues described earlier.

The other key issue with this scheme arises when a node with no data to send is polled. Besides wasting airtime, these 'idle polls' also lead to increased access delays for subsequent nodes that do have data to send. Several schemes have been proposed that pause the polling of such nodes if they remain idle for a number of polling rounds [39]-[41]. However, these approaches can cause "post-idle" delays of one or more polling rounds if the node resumes to have data to send before the pause duration ends. To address this issue, rather than pausing idle polls, some approaches [42], [43] move them to the end of the round. This allows nodes that resume having data to send before the end of the pause duration to transmit their data within the same round, even if it is later in the round. In the TAS context, an analogous approach would be to schedule the idle polls during the gaps between gating windows. Other approaches suggest switching to random access schemes [44], [45] to reduce this post-idle delay.

Other proposals addressing the idle poll issue include sending requests for polls, such as through short TDMA slots before each polling cycle [46] or Code Division Multiple Access (CDMA) [47]. However, these methods add overhead if not piggybacked on other data or control frames. Also, these requests must be sent well in advance of real-time data frames to avoid aggravating access delays. Thus, these schemes may benefit only traffic flows with known or predictable idle/busy periods, such as real-time video streaming [48].

D. *Contention-Based CSMA Issues & Related Works*

Contention-based CSMA was first proposed in [49]. Its most popular form is CSMA with Collision Avoidance (CSMA/CA) which is used in Wi-Fi [35]. In this scheme, nodes transmit if the channel is idle for a short sensing duration; otherwise, they defer randomly. This random deferment and the resulting collisions hinder low-latency operations and have been the focus of several studies. For instance, in [50], packets are deferred to a virtual time corresponding to their actual generation times. The virtual time advances only when the channel is idle, moving at Rv times the speed of real time, but once it reaches the real time, it follows the real time exactly. In this way, collisions are avoided if the actual generation times differ by at least Rv times the sensing duration. Switching to a contention-free protocol is also used to prevent collisions, where in [51] contention-based CSMA switches to contention-free CSMA when collisions occur, whereas in [52], a node uses the Request-To-Send/Clear-To-Send (RTS/CTS) protocol to reserve recurring TDMA slots, which other nodes avoid by maintaining a history of the channel's idle/busy periods.

E. *Contention-Free CSMA & Hyperchannel*

Contention-free CSMA protocols are rare, with only the Hyperchannel [7] adopted for commercial use, while others remain largely unexplored in the literature [53], [54]. These protocols

IEEE Proof

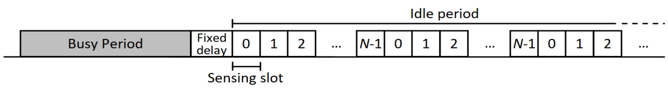

Fig. 2. Hyperchannel slotting sequence.

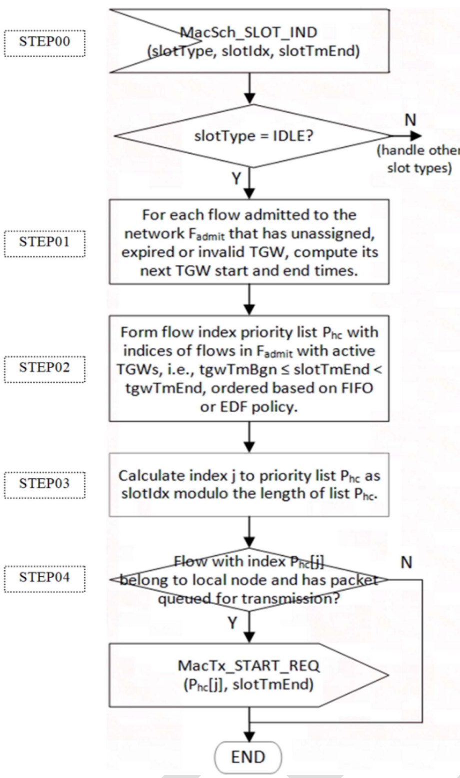

Fig. 3. Flowchart for TGT-HC's initial transmission CSMA routine.

involve nodes taking turns to sense the channel and transmit in a conflict-free sequence. For instance, in the Hyperchannel, after each busy period, the time is divided into idle slots which are recurringly numbered $\{0, 1, 2, \ldots, N-1\}$ (see Fig. 2). In each slot, a node with a matching index senses the channel. To support multiple flows per node, we can use flow indices $\{0, 1, 2, \ldots, F-1\}$ instead. The slot assignment order can then be expressed as a priority list $P_{hc} = \{0, 1, 2, \ldots, F-1\}$, where $P_{hc}[i] = i$. The Hyperchannel access scheme can then be expressed as an activation function φ that takes in P_{hc}, its length $|P_{hc}|$, a flow index f, and the number of idle slots s that have lapsed since the end of the busy period. The function returns 1 if a packet from flow f can be transmitted, else 0. Specifically,

$$\varphi_{hc}(P_{hc}, |P_{hc}|, f, s) = \begin{cases} 1 & \text{if } f = P_{hc}[(s-1) \bmod |P_{hc}|] \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

III. TGT-HC

A. Contention-Free CSMA Routine for Initial Transmission

Fig. 3 illustrates the CSMA routine for TGT-HC's initial transmission. The process begins when a slot indication signal is received (**STEP00**). If the slot is idle, then the next TGW start and end times for all flows admitted on the medium are calculated (**STEP01**). Each node must already have the flows' start times, cycle periods, and maximum allowed packet latencies, obtained from the flow admission scheme [3]. These parameters are stored in a flow descriptors list F_{admit} as the flow's first TGW start time $tgwTmBgn(0)$, cycle interval $tgwIntv$, and duration $tgwDur$. If a flow had started at $t = 0$, its current or next TGW start time at any subsequent time t would be given by

$$tgwTmBgn'(t) = \left\lfloor \frac{t + tgwIntv - tgwOfs - tgwDur - 1}{tgwIntv} \right\rfloor \times tgwIntv + tgwOfs \quad (2)$$

where $tgwOfs$ is the TGW offset relative to its interval:

$$tgwOfs = tgwTmBgn(0) \bmod tgwIntv \quad (3)$$

But since the flow actually starts at $tgwTmBgn(0)$, the TGW's start time is therefore:

$$tgwTmBgn(t) = max(tgwTmBgn'(t), tgwTmBgn(0)) \quad (4)$$

Meanwhile, the TGW's end time is simply:

$$tgwTmEnd(t) = tgwTmBgn(t) + tgwDur \quad (5)$$

Once all TGWs are computed,F_{admit} is sorted by TGW start/end time based on either FIFO or EDF policy (**STEP02**). The priority list P_{hc} is then created by selecting only the indices of flows with TGWs active at the slot's end time, i.e.,

$$tgwTmBgn(slotTmEnd) \leq slotTmEnd < tgwTmEnd(slotTmEnd) \quad (6)$$

Subsequently, a queued packet from flow f is transmitted at the slot's end time if $f = P_{hc}[j]$, where j is the remainder of the idle slot index divided by the length of P_{hc} (**STEP03-04**):

$$j = slotIdx \bmod |P_{hc}| \quad (7)$$

It is important to note that $tgwDur$ must be less than $tgwIntv$ for proper operation, but also longer than the maximum retransmissions duration $maxRetryDur$ to meet the reliability guarantee, i.e.,

$$maxRetryDur \leq tgwDur < tgwIntv \quad (8)$$

where $maxRetryDur$ is expressed as:

$$maxRetryDur = (1 + maxRetry).[datDur + sifDur + ackDur + sifDur] \quad (9)$$

where $datDur$ and $ackDur$ are the maximum data and acknowledgement frame durations, and $sifDur$ is the duration between the frames. Flows with maximum latency below $maxRetryDur$ should be rejected if their reliability guarantee cannot be met. It is also preferable to have $tgwDur$ closer to $maxRetryDur$ rather than $tgvIntv$ to reduce the overlaps between different flows' TGWs. This results in a shorter P_{hc} on average and, therefore, a reduced channel access delay. However, to prevent a previously overlapping flow from robbing the next flow's chance to transmit/retransmit sufficiently, the overlap should be added to $tgwDur$ as shown in Fig. 4.

B. Slot Sequencing and Retransmissions

Six types of slots or interframe spacings are defined in TGT-HC. The transmission (**TX**) slot spans the period when the PHY layer is transmitting a frame, while the reception (**RX**) slot is

IEEE Proof

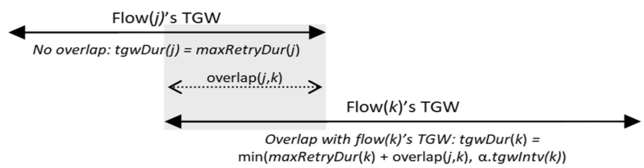

Fig. 4. Adjusting TGW duration for overlap between TGWs.

the period when it indicates the channel is busy, regardless of whether a frame is received. The Short Interframe Space (**SIFS**) and Distributed Interframe Space (**DIFS**) are based on IEEE 802.11's definitions, except that SIFS is added after each TX/RX slot, not just between the DATA and ACK frames. DIFS starts after SIFS rather than a busy period, and is skipped if the previous TX/RX slot is an ACK frame, identified by its shorter duration compared to the shortest DATA frame.

After DIFS comes the 'next' (**NXT**) slot, during which the next active flow in the Phc can start transmitting once the current one finishes. However, this bypass of the CSMA routine for the initial transmission only occurs if the node owning the next flow can discern the current flow's index from one of its DATA or ACK frames. Following the NXT slot are the **IDLE** slots, during which the CSMA routine arbitrates the next flow to be transmitted.

Fig. 5 shows an example of the slot and interframe spacing sequence for initial transmission and retransmissions. In TGT-HC, a node retransmits a SIFS and a DIFS after the previous attempt if the recipient does not send an ACK. If the ACK is sent but merely sensed or received corrupted, the node retransmits a SIFS after receiving the ACK. The DIFS is set to ACK + SIFS so that, regardless whether the ACK is sent, the next attempt starts at the same time relative to the previous.

The sequence starts with no active TGWs, so P_{hc} is null, and the first idle slot has no index. Shortly after, flow5's TGW starts, making P_{hc} become $\{5\}$, and the next slot is indexed as '5'. With flow5's packet already queued, the owning node transmits it at $\boldsymbol{t = 40\ \mu}$s after the slot is sensed idle. The node then waits for a SIFS and a DIFS. As no ACK is received, the DIFS elapses, and the packet is retransmitted at $\boldsymbol{t = 200\ \mu}$s. This time, the ACK is received, confirming packet delivery.

A SIFS later, at $\boldsymbol{t = 360\ \mu}$s, a NXT slot is entered because the previous busy period was shorter than the minimum DATA duration. As P_{hc} is still $\{5\}$, the following idle slot is indexed '5'. After this slot, flow1's TGW starts, so P_{hc} becomes $\{5, 1\}$, and thereafter the idle slot indices cycle within $\{5, 1\}$. At $\boldsymbol{t = 460\ \mu}$s, flow1's packet is transmitted as soon as the next available idle slot indexed '1' ends. A SIFS later, an ACK is received, confirming flow 1's packet delivery. At $\boldsymbol{t = 620\ \mu}$s, another NXT slot bypasses the DIFS. At this point, however, flow3 is next to flow1 in P_{hc} (now $\{5, 1, 3\}$), so flow3's packet is transmitted right after the slot, skipping the CSMA routine. A SIFS, an ACK and another SIFS later, another NXT slot is entered. Now, $P_{hc} = \{1, 3\}$, and the subsequent idle slot indices cycle within $\{1, 3\}$ since flow5's TGW has ended. Eventually, flow3 and then flow1's TGWs also end, reducing P_{hc} and the idle slot indices sequence to $\{1\}$, and finally to null.

C. Complexity Consideration

In TGT-HC, the flow sorting step (**STEP2** in Section III-A) is the most time-consuming, with a time complexity of $O(f\log f)$, where f is the number of flows. The other steps are comparisons and arithmetic operations with less significant time complexity terms of $O(f)$. This is not much different from TDMA or polling MAC if they also need to sort flows based on TGWs to support the TAS operation. In terms of space complexity, all three schemes would require $O(f)$ to keep track of all the flows' parameters.

The key difference is however in where and how frequently the algorithm is invoked: For TGT-HC, every node in the network invokes the computation at the start of each idle slot. For TDMA and the polling MAC, only the central controller needs to sort the flows, likely at the start of the superframe or a new polling turn, respectively. In this regard, TGT-HC is at a disadvantage if deployed on resource-constrained wireless devices. To alleviate this, it may be possible to limit TGT-HC's sorting step to after each flow has completed its transmission.

Nonetheless, as TGT-HC is based on contention-free CSMA, it has significant advantages over the other two schemes with its shorter access delay and less control overhead. Furthermore, as we shall show in Sections V-B and VI-C, its latency is significantly better compared to the TDMA-based RT-WiFi and the polling-MAC based IEEE 802.11ax.

IV. Analytical Evaluation

The single server FCFS and the TGT-HC MAC protocols are modelled by a basic M/G/1/J/M queue plus a vacation. The former MAC employs a modified exhaustive service for its vacation model from the M/G/1 queue, while the latter employs a modified gated service from the M/G/1 queue. The general service, G, can model a deterministic service for a busy period when the FER is zero or a truncated geometric service when it is greater than zero with transmission retries.

A. Analysis of the TGT-HC MAC Protocol With FER = 0%

The TGT-HC MAC protocol with zero FER is modelled as the above queue with a slot countdown delay. Its response delay, $T_{Response}$, is given by [55]

$$T_{Response} = \frac{M\bar{Z}}{1 - Q_0} - \frac{1}{\theta} - \frac{Q_J(M - J)\bar{Z}}{1 - Q_0}, \tag{10}$$

where J is the number of packet buffers, M is the number of stations/population sources, θ is a station's traffic arrival rate, $\bar{Z}$ is the average service period and $\bar{Z}^2$ is the second moment of service period. Denoting A as a station's traffic generation period, a station's arrival rate is given by

$$\theta = \frac{1}{A}. \tag{11}$$

$M\theta$ is the network's total arrival rate. The average service period is the busy period for transmitting a data and ACK frames with their Short Interframe Spaces (SIFSs). Denoting D_{Busy} as its busy period, we have

$$\bar{Z} = D_{Busy} = D_D + D_S + D_A + D_S, \tag{12}$$

IEEE Proof

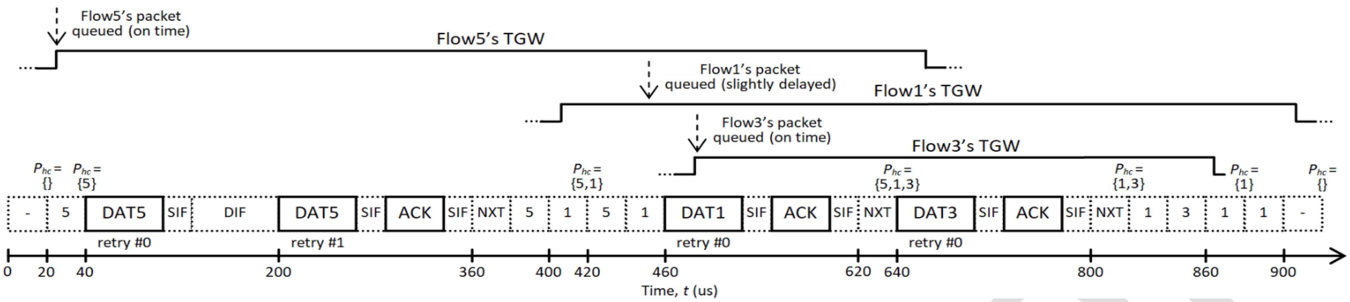

Fig. 5. TGT-HC slot and retransmission sequence.

where D_D is the data frame time, D_A is the ACK frame time and D_S is the SIFS time. Q_j is the probability of j data packet(s) in the network, and $j = 0, 1, 2, \ldots, J$. It is given by

$$Q_j = \frac{M\Omega_j}{(M-j)(\Omega_0 + M\theta\bar{Z})}, j = 0, 1, 2, \ldots, J-1, \quad (13)$$

and

$$Q_J = 1 - \sum_{j=0}^{J-1} Q_j. \quad (14)$$

Ω_j is given by

$$\Omega_j = \frac{\Omega_j^M}{\sum_{i=0}^{J-1}\Omega_i^M}, j = 0, 1, 2, \ldots, J-1, \quad (15)$$

Ω_j^M is the distribution of queue length and it is given by

$$\Omega_j^M = \sum_{m=M-j-1}^{M-1} D_m \binom{m}{M-j-1} (-1)^{m-(M-j-1)},$$
$$j = 0, 1, 2, \ldots, M-1, \quad (16)$$

$$D_j = \frac{H_j \sum_{m=j}^{M-1} \frac{\binom{M-1}{m}}{H_m}}{\sum_{m=0}^{M-1} \frac{\binom{M-1}{m}}{H_m}}, j = 0, 1, 2, \ldots, M-1, \quad (17)$$

$$H_0 = 1, \quad (18)$$

$$H_m = \prod_{k=1}^{m} Z^*(k\theta)/(1 - Z^*(k\theta)), m \geq 1, \quad (19)$$

where $Z^*(s = k\theta)$ is the Laplace-Stieltjes Transform of each of the service period distribution function given by

$$Z^*(s = k\theta) = e^{-s\bar{Z}} = e^{-k\theta\bar{Z}}. \quad (20)$$

Futhermore, the average slot countdown prior to sending a data frame, denoted by D_{SlotCD}, is given by a modification for an M/G/1 queue having a vacation model with gated service as [56]

$$D_{SlotCD} = \bar{V}(M + 2 - M\theta(1 - P_B)\bar{Z})$$
$$/2(1 - M\theta(1 - P_B)\bar{Z}), \quad (21)$$

where

$$\bar{V} = \begin{cases} D_{Slot}, & \text{if } M = 1, \\ \frac{[(M-1)q+1]D_{Slot}}{M}, & \text{if } M > 1, \end{cases} \quad (22)$$

D_{Slot} is a time slot, P_B is the blocking probability of the M/G/1/J/M queue and is given by

$$P_B = \frac{\theta\bar{Z}(M-J)Q_j}{1 - Q_0 + \theta\bar{Z}(M-J)Q_j}, \quad (23)$$

and q is given by

$$q = \frac{D_{VulDur}}{A}, \quad (24)$$

where D_{VulDur} is the vulnerable duration when a tagged data frame is being sent. It is dependent on the number of data frames in the queue before it is being sent. Thus, it is as follows.

$$D_{VulDur} = \begin{cases} \bar{J}\bar{Z}(1 - Q_0 - Q_1), & \text{if} \overline{JY}(1 - Q_0 - Q_1) < D_m, \\ D_m, & \text{if} \overline{JY}(1 - Q_0 - Q_1) \geq D_m, \end{cases} \quad (25)$$

where

$$\bar{J} = \sum_{j=0}^{J-1} jQ_j. \quad (26)$$

The probabilities Q_0 and Q_1 are subtracted from a probability of one as the system has zero and one packet (tagged packet) in the queue, causing no vulnerable duration. D_m is the maximum period for sending the data and ACK frames. In addition, note that the term $(M-1)q$ is due to the average vulnerable duration for a Binomial distribution Bino$(M-1, q)$. The delay variance of the slot countdown, denoted by D^v_{SlotCD}, is given by

$$D^v_{SlotCD} = (\overline{V^2} - \overline{V}^2)/2\overline{V}, \quad (27)$$

and

$$\overline{V^2} = \begin{cases} D^2_{Slot}, & \text{if } M=1, \\ \frac{\left[(M-1)^2\left(\frac{\overline{D^2_{VulDur}}}{A^2}\right) + 2(M-1)\left(\frac{D_{VulDur}}{A}\right) + 1\right] D^2_{Slot}}{M^2}, & \text{if } M>1. \end{cases} \quad (28)$$

$$\overline{D^2_{VulDur}} = \begin{cases} \overline{J^2}\,\overline{Z^2}(1 - Q_0 - Q_1)^2, & \text{if } \overline{JZ}(1 - Q_0 - Q_1) < D_m, \\ D^2_m, & \text{if } \overline{JZ}(1 - Q_0 - Q_1) \geq D_m, \end{cases} \quad (29)$$

where the average service period's second moment is given by

$$\overline{Z^2} = D^2_{Busy}, \quad (30)$$

IEEE Proof

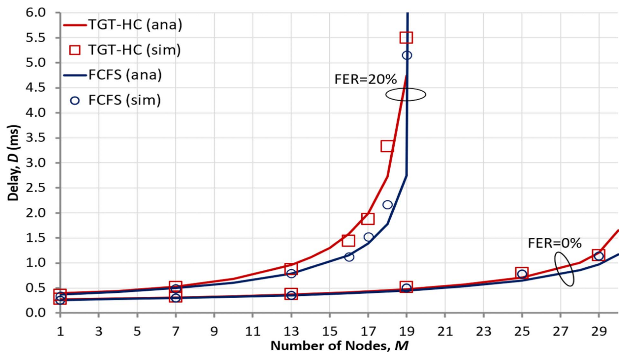

Fig. 6. Analytical versus simulation mean delays for TGT-HC and single-server FCFS at 0% and 20 FER.

and

$$\overline{J^2} = \sum_{j=0}^{J-1} j^2 Q_j. \quad (31)$$

The modification of the gated vacation model seems reasonable as shown in the simulation and analytical results in Fig. 6. Therefore, the total response delay of the TGT-HC MAC protocol, denoted by $D_{TotalResponse}$, is given by

$$D_{TotalResponse} = D_{Response} + D^{v}_{SlotCD} + D_{SlotCD}. \quad (32)$$

The real response delay of the system is measured at the moment that the data frame is correctly received without waiting for an ACK frame confirmation. Therefore, the average measured delay of the TGT-HC MAC protocol with zero FER, denoted by D_{TGTHC}, is given by

$$D_{TGTHC} = D_{TotalResponse} - D_{ACK\_to}. \quad (33)$$

where the ACK frame's timeout period D_{ACK_to} is given by

$$D_{ACK\_to} = D_S + D_A. \quad (34)$$

B. Analysis of Single Server FCFS Queue With $FER = 0\%$

The Single Server FCFS queue with zero FER is modelled using the same queue with a SIFS slot duration delay. The response delay can be determined from (10). Now, the average SIFS slot countdown before transmitting a data frame, denoted by D_{SCD} is given by a modification of a vacation model with an exhaustive service of an M/G/1 queue [56] as

$$D_{SCD} = \overline{U}(M - M\theta(1-P_B)\bar{Z})/2(1 - M\theta(1-P_B)\bar{Z}), \quad (35)$$

$$\overline{U} = \begin{cases} 0, & \text{if } M = 1, \\ \frac{[(M-1)r+1]D_S}{M}, & \text{if } M > 1, \end{cases} \quad (36)$$

and r is given by

$$r = \frac{D_{VulDur2}}{A}, \quad (37)$$

where $D_{VulDur2}$ is the vulnerable duration when a tagged data frame is being sent for the FCFS queue. It is dependent on the number of data frames in the queue before it is being sent. Thus, it is as follows.

$$D_{VulDur2} = \begin{cases} \bar{J}\bar{Z}(1-Q_0-Q_1), & \text{if } \bar{J}\bar{Y}(1-Q_0-Q_1) < A, \\ A, & \text{if } \bar{J}\bar{Y}(1-Q_0-Q_1) \geq A. \end{cases} \quad (38)$$

The delay variance of the SIFS slot countdown, denoted by D^{v}_{SCD}, is given by [56] as

$$D^{v}_{SCD} = (\overline{U^2} - \overline{U}^2)/2\overline{U}, \quad (39)$$

where

$$\overline{U^2} = \begin{cases} 0, & \text{if } M = 1, \\ \frac{\left[(M-1)^2\left(\frac{\overline{D^2_{VulDur2}}}{A^2}\right) + 2(M-1)\left(\frac{D_{VulDur2}}{A}\right) + 1\right] D^2_S}{M^2}, & \text{if } M > 1. \end{cases} \quad (40)$$

and

$$\overline{D^2_{VulDur2}} = \begin{cases} \overline{J^2}\,\overline{Z^2}(1-Q_0-Q_1)^2, & \text{if } \bar{J}\bar{Z}(1-Q_0-Q_1) < A, \\ A^2, & \text{if } \bar{J}\bar{Z}(1-Q_0-Q_1) \geq A. \end{cases} \quad (41)$$

The modification of the exhaustive vacation model is reasonable as shown in the simulation and analytical results of Fig. 6. Therefore, the total response delay of the Single Server FCFS queue, denoted by $D_{TotalResonse,SSFCFS}$, is given by

$$D_{TotalResponse,SSFCFS} = D_{Response} + D^{v}_{SCD} + D_{SCD}. \quad (42)$$

Therefore, the average measured delay of this queue with zero FER, denoted by $D_{AverageMeasured,SSFCFS}$, is given by

$$D_{AverageMeasured,SSFCFS} = D_{TotalResponse,SSFCFS} - T_{ACK\_to}. \quad (43)$$

C. Analysis of the TGT-HC MAC Protocol With FER > 0%

Next, the TGT-HC MAC protocol with FER $> 0\%$ is modelled using the same queue plus a slot countdown delay again. The data frame is re-sent when there is an error in the data frame and ACK frame. The maximum number of retries is denoted as N_R. The effective frame error rate for both the data and ACK frames, denoted by FER_1, is given by

$$FER_1 = 1-(1-FER)(1-FER). \quad (44)$$

where FER is the frame error rate of the data or ACK frame. The average service period is given by

$$\bar{Z} = \frac{\sum_{i=0}^{N_R} FER_1^i(1-FER_1)(iD_{Busy\_ns} + D_{Busy})}{\sum_{i=0}^{N_R} FER_1^i(1-FER_1)}, \quad (45)$$

where D_{Busy_ns} $(= D_{Busy} + D_{NXT})$ is the busy period when the data frame is not successfully transmitted and needs to retry again if its retry limit has not been exceeded. Furthermore, the second moment of the average service period is given by

$$\overline{Z^2} = \frac{\sum_{i=0}^{N_R} FER_1^i(1-FER_1)(iD_{Busy\_ns} + D_{Busy})^2}{\sum_{i=0}^{N_R} FER_1^i(1-FER_1)}. \quad (46)$$

The delay is measured after receiving the first data frame correctly without waiting for the ACK frame timeout. This results

IEEE Proof

TABLE I
ANALYTICAL EVALUATION SETTINGS

Parameter	**Value**
IDLE/NXT slot duration, D_{Slot}	20 μs
SIFS duration, D_S	10 μs
DATA frame duration, D_D	250 μs
ACK frame duration, D_A	50 μs
DIFS duration (ACK+SIFS)	60 μs
Busy period, D_{Busy}	320 μs
Busy period (unsuccessful), D_{Busy_ns}	320 μs
Maximum number of retries, N_R	7
Frame error rate, FER	$\{0, 20\}\%$
Packet interval, A	10 ms
Maximum latency ($tgwDur$), D_M	5 ms
Number of packet buffers, J	$M-1$

TABLE II
SIMULATION AND PROTOTYPE EVALUATION SETTINGS

Parameter	**Value**
Waveform	802.11g (20 MHz)
Modulation/coding	QPSK½ (12 Mbps)
IDLE/NXT slot duration	20 μs
SIFS duration	10 μs
DATA frame duration	250 μs
ACK frame duration	50 μs
DIFS duration (ACK+SIFS)	60 μs
Maximum retries	7
FER (emulated)	$\{0, 20\}\%$
Number of flows	$\{1, 4, 7, \ldots, 30\}$
Packet length	256 bytes
Packet interval	10 ms
Offset	Random within $\{0, 10\}$ ms
Maximum latency ($tgwDur$)	5 ms

in saving of two types of additional delays. The first type of delay is due to the saving of the ACK timeout duration and the second type of delay is due to first receiving the correct data frame without the correctly received ACK frame. For the first type of delay, denoted by D_1, it is given by

$$D_1 = \frac{\sum_{i=0}^{N_R} FER_1^i (1-FER_1)(D_{ACK\_to})}{\sum_{i=0}^{N_R} FER_1^i (1-FER_1)}, \tag{47}$$

where FER_2 is the effective frame error rate of a successfully received data frame and an unsuccessful ACK frame. FER_2 is given by

$$FER_2 = 1-(1-FER)FER. \tag{48}$$

Next, we consider the second type of delay. The additional ACK timeouts at the ith retry are possible only if both the data and ACK frames are successful at jth retry, where $j = 0, 1, 2, \ldots, N_R$, and $i = 0, 1, 2, \ldots, j$. In addition, there are also associated not successful busy periods that come with the ACK timeouts. These additional delays at the ith retry is possible only if both the data and ACK frames are successful at jth retry, where $j = 0, 1, 2, \ldots, N_R$, and $i = 1, 2, \ldots, j$. Thus, the second type of delay, denoted by D_2, is given by

$$\begin{aligned} D_2 &= \frac{\sum_{j=0}^{N_R} \sum_{i=0}^{j} FER_1^{j+i} (1-FER_1)(1-FER_2)(D_{ACK\_to})}{\sum_{j=0}^{N_R} \sum_{i=0}^{N_R} FER_1^{j+i} (1-FER_1)(1-FER_2)} \\ &+ \frac{\sum_{j=0}^{N_R} \sum_{i=1}^{j} FER_1^{j+i} (1-FER_1)(1-FER_2)(D_{Busy\_ns})}{\sum_{j=0}^{N_R} \sum_{i=0}^{N_R} FER_1^{j+i} (1-FER_1)(1-FER_2)}. \end{aligned} \tag{49}$$

Note that the average service period and the Laplace-Stieltjes transform of the distribution function of the service period cater for the data frame retries due to errors. The average service period is given by (45) and the Laplace-Stieltjes transform of the distribution function of the service period is given by

$$Z^*(s=k\theta) = \frac{\sum_{i=0}^{N_R} FER_1^i (1-FER_1) e^{-k\theta(iD_{Busy\_ns}+D_{Busy})}}{\sum_{i=0}^{N_R} FER_1^i (1-FER_1)} \tag{50}$$

By approximating the two types of busy periods to be of the same duration, we can determine the average number of busy periods. The mean number of busy period can be used to determine the new vulnerable duration. We will exclude the tagged data frame of a node first. Then, there is $(M-1)$ number of nodes that can affect the vulnerable duration of the tagged data frame. Let N_{Busy} denote the average number of busy periods from the angle of the tagged data frame. By modelling the average number of busy periods as the sum of $(M-1)$ geometric distributions, we have

$$N_{Busy} = \frac{\sum_{i=1}^{(M-1)(N_R+1)} \binom{i-1}{M-2} FER_1^{i-(M-1)} (1-FER_1)^{M-1} i}{\sum_{i=1}^{(M-1)(N_R+1)} \binom{i-1}{M-2} FER_1^{i-(M-1)} (1-FER_1)^{M-1}}. \tag{51}$$

The new vulnerable duration and its second moment are given by equations (52)–(53), respectively shown at the bottom of the next page.

The second moment of the number of busy periods is given by

$$\overline{N_{Busy}^2} = \frac{\sum_{i=1}^{(M-1)(N_R+1)} \binom{i-1}{M-2} FER_1^{i-(M-1)} (1-FER_1)^{M-1} i^2}{\sum_{i=1}^{(M-1)(N_R+1)} \binom{i-1}{M-2} FER_1^{i-(M-1)} (1-FER_1)^{M-1}}. \tag{54}$$

The response time is given by (10), the average slot countdown before transmitting a data frame is given by (21), its variance is given by (27) and their sum is given by (32). Finally, the measured delay of the TGT-HC MAC protocol with $FER > 0\%$, denoted by D_{TGTHC}, is given by

$$D_{TGTHC} = D_{TotalResponse} - D_1 - D_2. \tag{55}$$

D. Analysis of Single Server FCFS Queue With $FER > 0\%$

Finally, we model the Single Server FCFS queue with $FER > 0\%$ as an M/G/1/J/M queue plus an additional delay due to the first slot time duration. The analysis is the same as that for the TGT-HC MAC protocol with $FER > 0\%$, except that the delays due to the slot countdown is a SIFS slot time duration if the node is transmitting as in the analysis for the Single Server FCFS Queue with $FER = 0\%$. Furthermore, the delay due to retries of unsuccessful packets up to its maximum retry limit also applies here. Therefore, (35) to (42) and (44) to (50) also apply here.

The new vulnerable duration and its second moment are given by equations (56)–(57), respectively, at the bottom of the next page.

IEEE Proof

Therefore, the average measured delay of the Single Server FCFS queue with $FER > 0\%$, $D_{AverageMeasured,SSFCFS}$, is given by

$$D_{AverageMeasured,SSFCFS} = D_{TotalResponse,SSFCFS} - D_1 - D_2. \quad (58)$$

E. Numerical Results

Table I shows the parameters used in the simulation and analytical results. The simulation results were obtained from the OMNeT simulation described in the next section, while the analytical results are obtained from the preceding sections. Fig. 6 shows the average delays of the simulation and analytical results of the single server FCFS queue and the TGT-HC MAC protocol with $FER = 0\%$ and $FER = 20\%$, respectively. The numerical results show that they match well over most ranges of the number of nodes. The numerical results show that very low latency can be achieved using the TGT-HC MAC protocol that approaches the single server first come first serve queue.

V. Simulative Evaluation

A. Simulation Platform, Implementation and Test Setup

We pitted TGT-HC on OMNET INET4 [57] against several protocols: Hyperchannel [7], Assigned Slot Listen Before Talk (ASLBT) [53], Mini-Slotted Access Protocol (MSAP) [54], CSMA with transmission guiding tracks (CSMA-TGT) [58], Real-Time Wi-Fi (RT-WiFi) [23], IEEE 802.11 Distributed Coordination Function (DCF) [35] and the FCFS. While DCF is inbuilt with INET4, we implemented the other protocols by replacing DCF's Contention class with codes we wrote based on the references. For RT-WiFi, we used the out-of-slot retry mode since the number of maximum retries for the in-slot mode was not specified. For the FCFS scheme, a global list was used to track the current and next flows to be transmitted, based on packet generation times. This allows the node with the next flow to start transmitting it immediately after the current one finishes, without any channel access arbitration. However, SIFS timings were observed to account for PHY transmit/receive turnaround times.

In the simulation, up to 30 wireless nodes were randomly placed in a 300 × 300 m area, with an access point (AP) located at one edge. Each node transmitted a flow of 256-byte packets every 10 ms to an industrial controller connected to the AP via 1Gbps Ethernet. Received DATA/ACK frames were randomly corrupted based on a FER parameter. Transmit powers were set high enough (30 mW) to ensure no errors occurred at FER = 0% . Table II provides the settings used.

The simulation parameters were varied within 1,4,7...30 nodes and 0, 20% FER. For each setting, 30 trials were conducted using different random generator seeds for node placement and flow start times. Each trial continued until 100,000 packets were transmitted, collecting end-to-end delays, retry counts, packet losses and queue length statistics.

B. Simulation Results

Fig. 7 charts the 99.7th percentile end-to-end packet delay results from the simulations. It shows that TGT-HC's performance is closest to the FCFS scheme's, even matching it at 0% FER. A closer look reveals that, at 0% FER, TGT-HC's delay of 2.7 ms for 25 nodes outperforms its closest rival, MSAP, which has a delay of 4.6 ms. At 20% FER, TGT-HC's delay also remains significantly lower than MSAP's: 5.5 ms versus 7.6 ms. In terms of the number of supported flows, both TGT-HC and MSAP support 29 flows at 0% FER, surpassing conventional Wi-Fi (18) and RT-WiFi (22). At 20% FER, TGT-HC's supports 16 flows, just one fewer than MSAP (17), but still far exceeding RT-WiFi (7). For perspective, note that Wi-Fi could not support any flows at all at this high FER.

VI. Prototype Evaluation

A. Prototype Platform, Implementation and Test Setup

For hardware prototyping, we used National Instruments' (NI) Labview Next Generation (NXG) and its IEEE 802.11a/ac library [59], the Xilinx Integrated Software Environment (ISE) Design Suite [60], and three NI USRP-2974 SDR transceivers [61]. Our custom code consists of a Labview user interface (UI) on the host PC, and VHDL code for the TGT-HC MAC. This MAC interacts with NI's 802.11a/ac PHY/SDR and Host-FPGA libraries on the USRP (Fig. 8). Within the TGT-HC MAC are a clock drift and time synchronization module (Tm), transmit/receive modules (Tx/Rx) for packet queuing and selection, frame header and cyclic redundancy check (CRC) processing, and a control module (Ctl) which implements the TGT-HC algorithm.

To avoid external interference, the USRP antenna ports were interconnected using RF cables, splitters, combiners and attenuators (Fig. 9). Initially, a noise generator was used to emulate

$$D_{VulDur} = \begin{cases} N_{Busy} D_{Busy\_ns} (1-Q_0-Q_1), & \text{if } N_{Busy} D_{Busy\_ns} (1-Q_0-Q_1) < D_m, \\ D_m, & \text{if } N_{Busy} D_{Busy\_ns} (1-Q_0-Q_1) \geq D_m. \end{cases} \quad (52)$$

$$\overline{D^2_{VulDur}} = \begin{cases} \overline{N^2_{Busy}} D^2_{Busy\_ns} (1-Q_0-Q_1)^2, & \text{if } N_{Busy} D_{Busy\_ns} (1-Q_0-Q_1) < D_m, \\ D^2_m, & \text{if } N_{Busy} D_{Busy\_ns} (1-Q_0-Q_1) \geq D_m. \end{cases} \quad (53)$$

$$D_{VulDur2} = \begin{cases} N_{Busy} D_{Busy\_ns} (1-Q_0-Q_1), & \text{if } N_{Busy} D_{Busy\_ns} (1-Q_0-Q_1) < A, \\ A, & \text{if } N_{Busy} D_{Busy\_ns} (1-Q_0-Q_1) \geq A. \end{cases} \quad (56)$$

$$\overline{D^2_{VulDur2}} = \begin{cases} \overline{N^2_{Busy}} D^2_{Busy\_ns} (1-Q_0-Q_1)^2, & \text{if } N_{Busy} D_{Busy\_ns} (1-Q_0-Q_1) < A, \\ A^2, & \text{if } N_{Busy} D_{Busy\_ns} (1-Q_0-Q_1) \geq A. \end{cases} \quad (57)$$

IEEE Proof

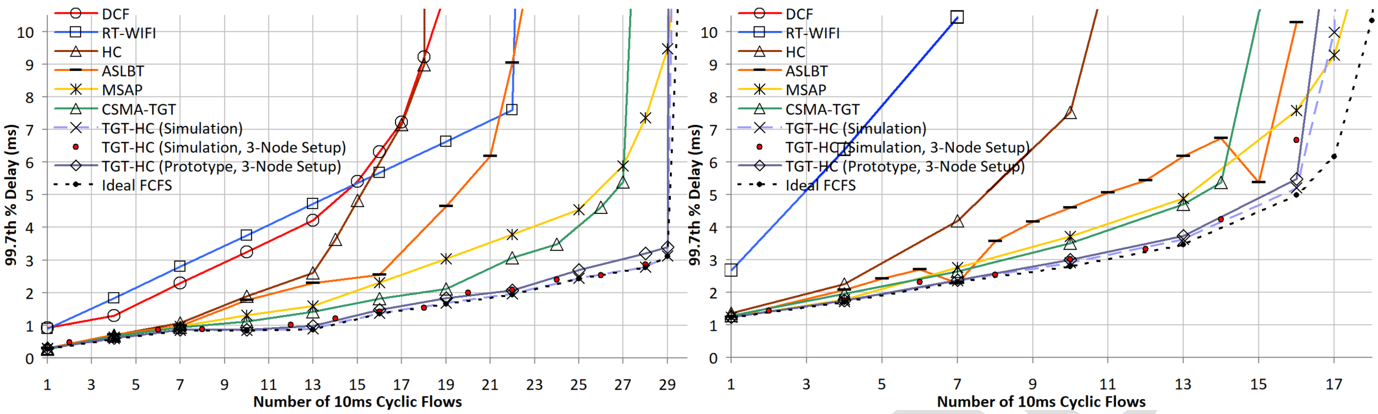

Fig. 7. 99.7th Percentile Packet Delays versus number of nodes for TGT-HC and other protocols at 0% (left) and 20% FER (right).

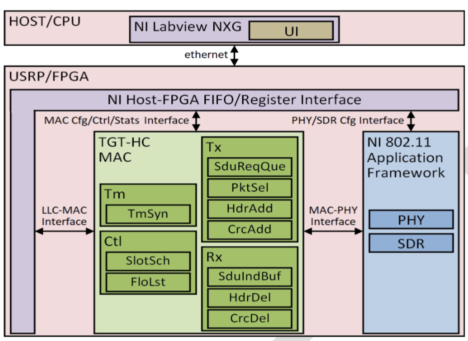

Fig. 8. TGT-HC hardware prototyping architecture.

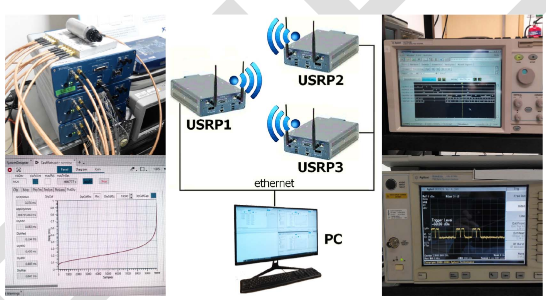

Fig. 9. Prototype evaluation setup.

TABLE III
OPTIMAL PROTOTYPE EVALUATION SETTINGS

Parameter	Value
Waveform	802.11a (20 MHz)
Modulation/coding	QPSK½ (12 Mbps)
IDLE/NXT slot duration	9 μs
SIFS duration	9 μs
DATA frame duration	87.65 μs
ACK frame duration	47.65 μs
DIFS duration (ACK+SIFS)	56.65 μs
Maximum retries	11
FER (emulated)	$\{0, 20\}\%$
Number of flows	$\{10, 20, 30\}$
Packet length	32 bytes
Packet interval	10 ms
Offset	Random within $\{0, 10\}$ ms
Maximum latency ($tgwDur$)	5 ms

frame errors, but the resulting FER values were difficult to control and varied too much across the nodes to match the settings used in the simulation. Therefore, we fell back to using the method adopted in the simulation where codes were added to randomly fail CRC checks at each node based on a FER setting on the UI. Transmit powers were set high enough to eliminate frame errors at FER $= 0\%$. We note that this method may not fully replicate the actual high FER scenario which may also include frame detection errors. However, we found that even at roughly 40% FER induced by an additive white gaussian noise (AWGN) noise generator, the USRP's frame detection rate remained exceptionally high at 99.99% , with almost all frame drops due to CRC failures.

Due to budget constraints, we only had three USRPs available. As the test setup for the prototype, we assigned two of the USRPs to alternately transmit up to 30 flows, based on the starting times of the flows' current or next TGWs, while the third USRP received and recorded the packet statistics. Although this setup differs from the simulation setup which involved up to 30 nodes each transmitting a flow, we found that in simulation, this 3-node setup yielded identical latency results to those from the single flow per node setup (Fig. 7). Additionally, our debug logs recorded the same slot sequences for both setups, confirming the accuracy of this 3-node setup in approximating the single flow per node setup.

As in the simulation, we varied the number of flows and FER settings, conducting 30 trials per setting with different random generator seeds for the flow start times. However, statistics were only collected for 33,334 packets per trial as the Labview UI became sluggish with larger data sets.

B. Prototype Lab Evaluation Results

We first validated the prototype against the simulation using the same test settings (Table II). Under these same conditions, the prototype achieved 99.7th percentile delays that closely matched those of the simulation (Fig. 7). We then tested with smaller packet sizes more typical of control applications, and reduced the interframe and slot durations (Table III). Fig. 10 shows the logCDF results obtained from the prototype and the simulated

IEEE Proof

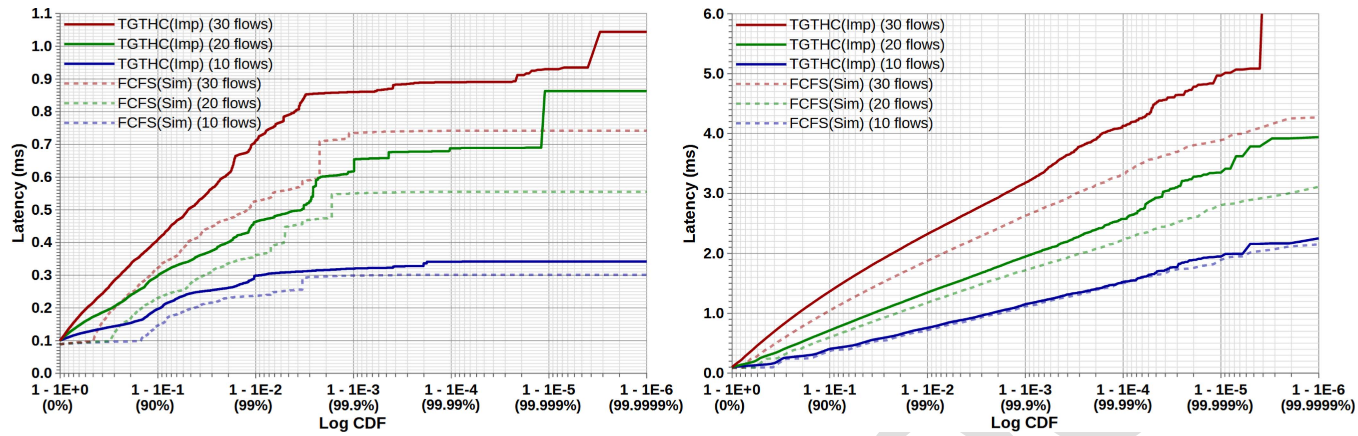

Fig. 10. Packet latency logCDFs for the TGT-HC prototype versus the FCFS simulation for 10, 20 and 30 flows of 32-byte packets at 10 ms interval, for 0% (left) and 20% FER (right).

TABLE IV
TGT-HC PROTOTYPE LATENCIES VERSUS IEEE 802.11AC/AX'S (20 MHZ BANDWIDTH)

Comparison	Subchannels	MCS	Channel Condition	Flows	Packet Size	Interval	Gurantee	Latency	Loss Rate
802.11ac (experiment) [19]	1	Adaptive	Office overnight 10m NLOS	1	100B	10ms	99.0%	3.0ms	0.015%
TGT-HC (prototype)		QPSK1/2	10% FER	1 / 16	256B		99.7%	0.9ms / 3.0ms	<0.0003%
802.11ac (theory) [18]	1	64QAM	Ideal	5	256B	Single	Unspecified	1.0ms	Unspecified
TGT-HC (prototype)		QPSK1/2	0% FER			10ms	99.7%	0.7ms	0%
802.11ax (theory) [18]	9	64QAM	Ideal	10	256B	Single	Unspecified	0.8ms	Unspecified
TGT-HC (prototype)	1	QPSK1/2	0% FER			10ms	99.7%	0.9ms	0%
802.11ax (simulation) [20]	9	Unspecified	TGn model E, 50m	9	100B	Single	99.999%	1.0ms	Unspecified
TGT-HC (prototype)	1	QPSK1/2	10% FER	30	32B	10ms	99.999%	1.2ms	<0.0003%

FCFS scheme based on these optimized settings. With 30 flows at 0% FER, the prototype's 99.999th percentile packet latency at 0.93 ms was somewhat higher than the FCFS's 0.74 ms. At 20% FER however, the figure rose to 5.0 ms, a full millisecond higher than the FCFS's 4.0 ms. With 10 flows, the results were much closer, with the prototype recording 0.34 ms and 1.9 ms at 0% and 20% FER respectively, compared to the FCFS's 0.30 ms and 1.8 ms. Packet loss rates for the prototype and FCFS were similar in all cases, recording 0% at 0% FER and approximately 0.0003% at 20% FER.

C. Comparison With TSN-Enabled 802.11ac/ax

For the same PHY bandwidth (20 MHz), our prototype outperformed or was at least comparable to the TSN-enabled Wi-Fi 6/7 in terms of latency [18], [19], [20] (Table IV). This is notable considering our prototype did not support simultaneous transmissions in multiple subchannels, and was only using Quadrature Phase Shift Keying with half coding rate (QPSK1/2) modulation and coding scheme (MCS). For instance, in [19], an actual IEEE 802.11ac station (STA) was set up to alternately transmit and receive a 100-byte packet every 10 ms to/from an AP 10 m away, without MCS limit, in a non-line-of-sight (NLOS) office environment overnight. Compared to this setup, TGT-HC at 10% FER achieved better latency (0.9 ms versus 3.0 ms) even with larger packets (256-byte) and lower MCS (QPSK1/2). For the same latency guarantee, it supported significantly more flows (16 versus 1).

In [18], the average latencies for IEEE 80211ac/ax STAs each transmitting a single 256-byte packet to an AP using 64-Quadrature Amplitude Modulation (64QAM) MCS under ideal channel conditions were derived theoretically. Compared to the five IEEE 802.11ac STAs setup, TGT-HC's latency for the same number of nodes was somewhat better (0.7 ms versus 1.0 ms) despite using a much lower MCS (QPSK1/2). When compared to the ten 802.11ax STAs setup which used nine subchannels for simultaneous transmissions, TGT-HC's latency for ten nodes was only slightly worse (0.9 ms versus 0.8 ms) despite using the much lower MCS and not utilising multiple subchannels simultaneously.

Finally, in [20], simulation results were obtained for IEEE 802.11ax STAs, each transmitting a 100-byte packet to an AP within a 50 m range using at least nine subchannels in a TGn E channel model. For similar throughput (30x32 versus 9x100 bytes/ cycle) with the same latency guarantee (99.999%), TGT-HC's latency was only slightly higher (1.2 ms versus 1.0 ms) at 10% FER despite not using multiple subchannels.

VII. CONCLUSION

In this paper, we presented a new TAS scheduler-driven contention-free CSMA scheme. Our analytical, simulation and prototype evaluations reveal its latency to be very close to that of the FCFS scheme, even at high FERs. Comparisons with TSN-enabled IEEE 802.11ac/ax also show promising despite its current limitations. Moving forward, we hope to have the

IEEE Proof

opportunity to upgrade it to support multiple subchannels, and integrate it into future TSN-enabled Wi-Fi and/or 5G/6G protocols, as well as validate its utility in actual deployments.

Acknowledgment

Dedicated to the first author's mother, Tan Gee Too. The second author thanks the late Prof. Jon W. Mark, Prof. Kin Mun Lye and Prof. Kee Chaing Chua for their queueing and network courses. Many thanks to National Instruments Singapore for technical support rendered during the project. Thanks also to Bo Jin, Yugang Ma, Leng Meng Goh and Wang Cho Cheng for their support during the project.

References

[1] E. Khorov, I. Levitsky, and I. F. Akyildiz, "Current status and directions of IEEE 802.11be, the future Wi-Fi 7," *IEEE Access*, vol. 8, pp. 88664–88688, 2020.
[2] R. Ali, Y. B. Zikria, A. K. Bashir, S. Garg, and H. S. Kim, "URLLC for 5G and beyond: Requirements, enabling incumbent technologies and network intelligence," *IEEE Access*, vol. 9, pp. 67064–67095, 2021.
[3] T. Adame, M. Carrascosa-Zamacois, and B. Bellalta, "Time-sensitive networking in IEEE 802.11be: On the way to low-latency WiFi 7," *Sensors*, vol. 21, no. 15, Jul. 2021, Art. no. 4954.
[4] O. Seijo, I. Val, and J. A. Lopez-Fernandez, "w-SHARP: Implementation of a high-performance wireless time-sensitive network for low latency and ultra-low cycle time industrial applications," *IEEE Trans. Ind. Inform.*, vol. 17, no. 5, pp. 3651–3662, May 2021.
[5] D. Cavalcanti, M. Rashid, L. Cariou, G. Venkatesan, and R. Stacey, "Controlling latency in 802.11," document IEEE 802.11-18/1160r0, Jul. 2018.
[6] X. Zuo et al., "Use of uplink persistent allocation for RTA," document IEEE 802.11-19/1538r1, Sep. 2019.
[7] I. Chlamtac and W. R. Franta, "Message-based priority access to local networks," *Comput. Commun.*, vol. 3, no. 2, pp. 77–84, Apr. 1980.
[8] *IEEE Standard for Local and Metropolitan Area Networks-Bridges and Bridged Networks-Amendment 25: Enhancements for Scheduled Traffic*, IEEE Standard 802.1Qbv-2015 (Amendment to IEEE Standard 802.1Q-2014), pp. 1–57, Mar., 2016.
[9] R. J. Jayabal et al., "TGT-HC: A time-aware shaper scheduled hyperchannel protocol for wireless time sensitive networks (TSNs)," in *Proc. 94th IEEE Veh. Technol. Conf.*, 2021, pp. 1–6.
[10] H. Kopetz and G. Grunsteidl, "TTP - A time-triggered protocol for fault-tolerant real-time systems," in *Proc. 23rd IEEE Int. Symp. Fault-Tolerant Comput.*, 1993, pp. 524–533.
[11] D. Krummacker and L. Wendling, "TSN simulation: Time-aware shaper implemented in NS-3," in *Proc. Workshop Next Gener. Netw. Appl.*, 2020, pp. 16–17.
[12] V. Eramo, T. Fiori, F. G. Lavacca, and F. Valente, "High-bandwidth technologies for next-generation launcher networks: A comparative analysis between TTethernet and time sensitive networking (TSN)," in *Proc. Int. Astronautical Congr.*, 2023, pp. 1–14.
[13] A. Nasrallah et al., "Performance comparison of IEEE 802.1 TSN time aware shaper (TAS) and asynchronous traffic shaper (ATS)," *IEEE Access*, vol. 7, pp. 44165–44181, 2019.
[14] S. Bhattacharjee, R. Schmidt, K. Katsalis, C. Y. Chang, T. Bauschert, and N. Nikaein, "Time-sensitive networking for 5G fronthaul networks," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–7.
[15] M. Barzegaran, N. Reusch, L. Zhao, S. S. Craciunas, and P. Pop, "Real-time traffic guarantees in heterogeneous time-sensitive networks," in *Proc. 30th Int. Conf. Real-Time Netw. Syst.*, 2022, pp. 46–57.
[16] X. Yuan et al., "A DQN-based frame aggregation and task offloading approach for edge-enabled IoMT," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 3, pp. 1339–1351, May/Jun. 2023.
[17] X. Yuan et al., "A MEC offloading strategy based on improved DQN and simulated annealing for Internet of Behavior," *ACM Trans. Sensor Netw.*, vol. 19, no. 2, pp. 1–20, Dec. 2022.
[18] D. Cavalcanti, J. Perez-Ramirez, M. M. Rashid, J. Fang, M. Galeev, and K. B. Stanton, "Extending accurate time distribution and timeliness capabilities over the air to enable future wireless industrial automation systems," in *Proc. IEEE*, vol. 107, no. 6, pp. 1132–1152, Jun. 2019.
[19] D. Cavalcanti, M. Rashid, G. Venkatesan, L. Cariou, and R. Stacey, "Time-aware shaping (802.1Qbv) support in the 802.11 MAC," document IEEE802.11-18/1892r0, Nov. 2018.
[20] D. Cavalcanti, G. Venkatesan, L. Cariou, and C. Cordeiro, "TSN support in 802.11 and potential extensions for TGbe," document IEEE 802.11-19/1287r1, Jul. 2019.
[21] W. Liang et al., "WIA-FA and its applications to digital factory: A wireless network solution for factory automation," in *Proc. IEEE*, vol. 107, no. 6, pp. 1053–1073, Jun. 2019.
[22] M. Rentschler et al., "IO-link wireless: The new standard for factory automation," in *Proc. 15th Wireless Congr.: Syst. Appl.*, 2018, pp. 14–15.
[23] Y. H. Wei, Q. Leng, S. Han, A. K. Mok, W. Zhang, and M. Tomizuka, "RT-WiFi: Real-time high-speed communication protocol for wireless cyber-physical control applications," in *Proc. IEEE 34th Real-Time Syst. Symp.*, 2013, pp. 140–149.
[24] M. Luvisotto, Z. Pang, and D. Dzung, "High-performance wireless networks for industrial control applications: New targets and feasibility," in *Proc. IEEE*, vol. 107, no. 6, pp. 1074–1093, Jun. 2019.
[25] A. Damnjanovic, W. Chen, S. Patel, Y. Xue, K. Hosseini, and J. Montojo, "Techniques for enabling low latency operation in LTE networks," in *Proc. IEEE Globecom Workshops*, 2016, pp. 1–7.
[26] K. I. Pedersen, G. Berardinelli, F. Frederiksen, P. Mogensen, and A. Szufarska, "A flexible 5G frame structure design for frequency-division duplex cases," *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 53–59, Mar. 2016.
[27] R. Costa, J. Lau, P. Portugal, F. Vasques, and R. Moraes, "Handling real-time communication in infrastructured IEEE 802.11 wireless networks: The RT-WiFi approach," *J. Commun. Netw.*, vol. 21, no. 3, pp. 319–334, Jun. 2019.
[28] X. Meng, J. Li, D. Zhou, and D. Yang, "5G technology requirements and related test environments for evaluation," *China Commun.*, vol. 13, pp. 42–51, 2016.
[29] S. A. Ashraf, I. Aktas, E. Eriksson, K. W. Helmersson, and J. Ansari, "Ultra-reliable and low-latency communication for wireless factory automation: From LTE to 5G," in *Proc. 21st IEEE Int. Conf. Emerg. Technol. Factory Automat.*, 2016, pp. 1–8.
[30] S. Nagata, L. H. Wang, and K. Takeda, "Industry perspectives," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 2–4, Jun. 2017.
[31] 3GPP, "NR; physical channels and modulation (release 15)," 3GPP TR 38.211, Dec. 2017.
[32] R. Abreu, P. Mogensen, and K. I. Pedersen, "Pre-scheduled resources for retransmissions in ultra-reliable and low latency communications," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2017, pp. 1–5.
[33] Y. Chen, H. Zhang, N. Fisher, and G. Yin, "Probabilistic per-packet real-time guarantees for wireless networked sensing and control," *IEEE Trans. Ind. Inform.*, vol. 14, no. 5, pp. 2133–2145, May 2018.
[34] D. J. Deng et al., "IEEE 802.11ax: Highly efficient WLANs for intelligent information infrastructure," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 52–59, Dec. 2017.
[35] *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2016 (Revision of IEEE Standard 802.11-2012), Dec., 2016.
[36] Z. Lu, Z. Khan, and M. A. Iqbal, "Fast polling mechanism for baseline BAN MAC (802.15. 6) of body area networks," *Procedia Comput. Sci.*, vol. 19, pp. 944–949, Jan. 2013.
[37] R. Y. Lam, V. C. Leung, and H. C. Chan, "Polling-based protocols for packet voice transport over IEEE 802.11 wireless local area networks," *IEEE Wireless Commun.*, vol. 13, no. 1, pp. 22–29, Feb. 2006.
[38] J. Lin, W. Liang, P. Zeng, H. Yu, and Y. Xiao, "Polling in the frequency domain: A new MAC protocol for industrial wireless network in factory automation," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 20, no. 4, pp. 211–22, Jan. 2015.
[39] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 116–126, Sep. 1997.
[40] Y. J. Kim and Y. J. Suh, "Adaptive polling MAC schemes for IEEE 802.11 wireless LANs supporting voice-over-IP (VoIP) services," *Wireless Commun. Mobile Comput.*, vol. 4, no. 8, pp. 903–916, Dec. 2004.
[41] J. Son, H. Choi, and S. C. Park, "An effective polling MAC scheme for IEEE 802. 11e," in *Proc. IEEE Int. Symp. Commun. Inf. Technol.*, 2004, pp. 296–301.
[42] J. Zheng and E. Regentova, "I-poll: Improved polling scheme for augmenting voice support in IEEE 802.11 PCF," *IEICE Trans. Commun.*, vol. 89, no. 6, pp. 1964–1967, Jun. 2006.

IEEE Proof

[43] H. V. Prabhu and G. S. Nagaraja, "Quality of service guaranteed delay sensitive polling algorithm for WiMax network: PQ_Poll," in *Proc. 2nd Int. Conf. Elect., Comput. Commun. Technol.*, 2017, pp. 1–6.
[44] A. Kopsel and A. Wolisz, "Voice transmission in an IEEE 802.11 WLAN based access network," in *Proc. 4th ACM Int. Workshop Wireless Mobile Multimedia*, 2001, pp. 24–33.
[45] E. Avdotin, D. Bankov, E. Khorov, and A. Lyakhov, "Enabling massive real-time applications in IEEE 802.11be networks," in *Proc. 30th IEEE Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2019, pp. 1–6.
[46] Y. Lv et al., "Request-based polling access: Investigation of novel wireless LAN MAC scheme for low-latency e-health applications," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 896–899, May 2019.
[47] J. H. Ko, S. Kwon, and C. Kim, "A polling method using orthogonal signalling for wireless local area networks," in *Proc. 34th IEEE Conf. Local Comput. Netw.*, 2009, pp. 130–136.
[48] M. A. Al-Maqri, M. Othman, B. M. Ali, and Z. B. Hanapi, "Feasible HCCA polling mechanism for video transmission in IEEE 802.11e WLANs," *Wireless Pers. Commun.*, vol. 85, no. 3, pp. 899–924, Dec. 2015.
[49] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part I-carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. 23, no. 12, pp. 1400–1416, Dec. 1975.
[50] M. L. Molle and L. Kleinrock, "Virtual time CSMA: Why two clocks are better than one," *IEEE Trans. Commun.*, vol. 33, no. 9, pp. 919–933, Sep. 1985.
[51] T. Nute, "HCSMA- a nearly conflict-free hybrid CSMA protocol with high utilization," in *Proc. IEEE Comput. Netw. Symp.*, 1984, pp. 75–84.
[52] S. Singh, P. A. K. Acharya, U. Madhow, and E. M. Belding-Royer, "Sticky CSMA/CA: Implicit synchronization and real-time QoS in mesh networks," *Ad Hoc Netw.*, vol. 5, no. 6, pp. 744–768, Aug. 2007.
[53] L. W. Hansen and M. Schwartz, "An assigned-slot listen-before-transmission protocol for a multi-access data channel," *IEEE Trans. Commun.*, vol. 27, no. 6, pp. 846–857, Jun. 1979.
[54] L. Kleinrock and M. Scholl, "Packet switching in radio channels: New conflict-free multiple access schemes," *IEEE Trans. Commun.*, vol. 28, no. 7, pp. 1015–1029, Jul. 1980.
[55] T. Takine, H. Takagi, and T. Hasegawa, "Analysis of an M/G/1/K/N queue," *J. Appl. Probability*, vol. 30, pp. 446–454, 1993.
[56] D. Bertsekas and R. G. Gallager, *Data Networks*, 2nd. ed. Englewood Cliffs, NJ, USA: Prentice Hall, 1992.
[57] OMNeT Community, "OMNeT: Discrete event simulation environment," Accessed: Aug. 2024. [Online]. Available: https://omnetpp.org/
[58] R. J. Jayabal et al., "CSMA-TGT: Carrier-sense multiple access with transmission guiding tracks," in *Proc. IEEE VTS Asia Pacific Wireless Commun. Symp.*, 2019, pp. 1–5.
[59] National Instruments, "Labview NXG," Accessed: Aug. 2024. [Online]. Available: https://www.ni.com/en/shop/labview/labview-nxg.html
[60] Xilinx Inc., "ISE design suite," Accessed: Aug. 2024. [Online]. Available: https://www.xilinx.com/products/design-tools/ise-design-suite.html
[61] National Instruments, "USRP-2974 high performance embedded SDR, 10 MHz-6GHz, 160MHz bandwidth," Accessed: Aug. 2024. [Online]. Available: https://www.ettus.com/all-products/usrp-2974/

Raymond J. Jayabal (Member, IEEE) received the BEng and MSc degree from University Malaya, Malaysia, and the PhD degree from Nanyang Technological University, Singapore. He is a Principal Scientist with the Institute for Infocomm Research, A*STAR, Singapore, specializing in time-sensitive MAC, hardware-in-the-loop simulation (HIL) and link scheduling optimization for ad hoc networks.

David Tung Chong Wong (Senior Member, IEEE) received the BEng and MEng degrees from the National University of Singapore, and the PhD degree from the University of Waterloo, Canada. His main research focus is on MAC protocols. He has served on numerous conference Technical Programme Committees (TPCs), been a Lead Editor for an MDPI journal special issue, and authored over 100 publications including journals, conference papers and book chapters. He has also published a book on Wireless Broadband Networks by John Wiley and Sons.

Lee Kee Goh received the BEng, MEng, and PhD degrees from the National University of Singapore. He is a Principal Research Engineer with the Communications and Networks Department of the Institute for Infocomm Research, A*STAR, Singapore. His research interests are in multiprocessor systems, parallel and distributed systems, energy-aware scheduling, and wireless networks.

Xiaojuan Zhang (Member, IEEE) received the BS degree in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2003 and the PhD degree from Nanyang Technological University, Singapore, in 2010. Currently, she serves as a Senior Scientist with the Institute for Infocomm Research, A*STAR, Singapore, specializing in Internet of Things (IoT), integrated sensing and communications (ISAC), MIMO systems and information theory.

Chin Ming Pang received the BEng degree from the National University of Singapore. He is a Senior Principal Research Engineer with the Communications & Networks Department of the Institute for Infocomm Research, A*STAR, Singapore. His work interests spans various areas in wireless communications technologies, FPGA hardware prototyping and embedded systems.

Sumei Sun (Fellow, IEEE) is Executive Director of the Institute for Infocomm Research, A*STAR, Singapore. She is a fellow of the Academy of Engineering, Singapore. Her research spans signal processing and optimization for wireless communications, currently focusing on Industrial Internet of Things (IoT), 5G/6G technologies and beyond technologies, and deep learning applications in wireless communications.

IEEE Proof