Removing the Wires in Time-Sensitive Networks

Pablo Avila-Campos
IDLab, Ghent University – imec
Ghent, Belgium
Email: pabloesteban.avilacampos@ugent.be

Jetmir Haxhibeqiri
IDLab, Ghent University – imec
Ghent, Belgium
Email: jetmir.haxhibeqiri@ugent.be

Xianjun Jiao
IDLab, Ghent University – imec
Ghent, Belgium
Email: xianjun.jiao@ugent.be

Ingrid Moerman
IDLab, Ghent University – imec
Ghent, Belgium
Email: ingrid.moerman@ugent.be

Jeroen Hoebeke
IDLab, Ghent University – imec
Ghent, Belgium
Email: jeroen.hoebeke@ugent.be

***Abstract*—From machine control applications to multimedia, and distributed real-time monitoring, all require a high level of determinism and low end-to-end latency. Time-Sensitive Networking, running on top of Ethernet, has been proposed as the solution over the past years. This paper analyses the crucial challenges that need to be overcome to go beyond Ethernet and seamlessly extend such time-sensitive operation into the wireless domain. A TSN-compatible design, based on our open-source WiFi implementation, is presented, with a specific focus on scheduling. Two experiments, in our IEEE802.11-based W-TSN testbed, comparing contention arising from a shared and dedicated time slot were performed. The results show a lower difference in the number of received packets between shared and dedicated time slots at smaller time slots.**

***Index Terms*—wireless time-sensitive networking, time synchronization, scheduling, openwifi, WiFi.**

I. Introduction

Rapid developments during the last decade in an increased machine and people inter-connectivity have paved the road for the deployment of diverse applications with ever-increasing demands. Such an increase in demands is not limited to data rate requirements only but also pertains to ultra-reliability and low latency deterministic communication. Such demanding applications span different verticals from industrial automation, audio/video applications, augmented/virtual reality, to automation and aerospace industry as shown in Table I.

To support such increasing demand for deterministic communication, a set of standards denoted as Time-Sensitive Networking (TSN) and applicable to Ethernet networks has been initiated by the IEEE 802.1 TSN task group. These standards define an extensive set of network features that deal with deterministic communication including accurate time synchronization [1], traffic scheduling and organization [2], traffic preemption [3], stream policing [4] and traffic duplication [5]. TSN configuration is standardized under IEEE 802.1Qcc [6], offering fully distributed, fully centralized or mixed (network centralized/ user distributed) configuration models assuming wire connections between all network nodes as well as end-devices.

Looking at the example applications in Table I, it is evident that many of them should not be confined to wired networks only, but require support for mobile devices (AR/VR applications, robotic/AGV communication, etc.) or portable devices (monitoring, control loops) to increase flexibility. Thus, extending TSN to wireless networks is a must to keep track of deterministic application requirements. In the past, wireless technologies that offered deterministic communication (WirelessHart, IEEE 802.15.4 TSCH, ISA 100.11a [7], [8]) have been defined, but the offered data rates were only in the range of hundreds of kBs. Looking at the wireless technology landscape, the most prominent candidates for higher data rate deterministic networking are the 3GPP's cellular networks (5G) and IEEE 802.11 (WiFi 6). While many times these two technologies are presented as competitors, for offering time-sensitive network features they should rather be looked at as complementary solutions. 5G operates in license bands and incurs higher deployment and operational costs [9], and can offer deterministic communication providing its ultra-reliable low latency communications (URRLC) set of features that can be integrated with wired TSN. In the case of 5G, the deployment types can be manyfold, being it a full stand-alone private deployment, a mixed deployment where the data traffic is confined within a private network while the control is done from the public network or a private network slice in a public network [10]. In cases of small to medium indoor deployments where communication needs to be confined to private networks only, WiFi-based TSN networks can prove to be feasible as well. WiFi has always been a natural extension of Ethernet networks and comes with far less management and operational complexity than 5G. However, in the past, as WiFi operates in an unlicensed spectrum, a lot of emphasis has been put on further increasing data rates, scaling the networks, and achieving fairness in accessing the wireless medium. However, with recent advances in WiFi specifications and promising research in adding TSN features, WiFi is positioning itself as a cost-efficient candidate for wireless TSN and will be the focus of this paper.

In this paper, we present a novel preliminary study measuring how the time slot size in a wireless TSN influences the cycle throughput. It compares dedicated and shared time slot cases showing the influence of contention.

Before diving into the road to WiFi TSN, section II will give a short introduction to TSN in general. Next, section IV details the requirements for a WiFi-based TSN and the network challenges in realizing it, while section V gives a preliminary assessment of scheduling strategies for uplink traffic in W-TSN. Section VI will conclude this paper with an outlook on future developments.

Table I
TIME-SENSITIVE NETWORKING USE CASE APPLICATIONS AND THEIR LATENCY REQUIREMENTS

	Applications	Latency(ms)
Industry 4.0	AGV/cranes	10 - 20
	Functional safety	1-10
	Condition monitoring	15-150
	Close loop control	0.5-5
	PLC/robot	0.5-2
Audio/Video/ Haptics	Live performance	0.5-5
	Presentation	5-10
	Tour guide	5-10
	Haptics	0.5-5
	AV conferencing	5-20
AR/VR	Entry-level WI	10-20
	Ultimate WI	1-10
	Entry-level SI	1-10
	Ultimate SI	1-5

II. TSN TUTORIAL

Time-Sensitive Networking (TSN) is a set of standards defined by the IEEE 802.1 TSN task group[1] to support deterministic communication on top of Ethernet-based (IEEE 802.3) networks. It consists of specifications for several network features: end-to-end accurate time synchronization [1], traffic scheduling and organization [2], [11], traffic policing [4], traffic duplication [5] as well as network and application configuration [6].

The smallest set of features needed to achieve deterministic networking is end-to-end accurate time synchronization and traffic scheduling. The IEEE802.1AS standard defines the mechanism of how the time is synchronized and maintained between all network devices. The time information is propagated from the time grandmaster to all other network devices. On each network hop, the time synchronization is based on a two-way exchange of the synchronization packets. *Sync* packets are propagated from the master across the link towards the slave and are followed by *Follow-up* packets for more accurate timestamping. To account for the link delays the time slave will respond with a *Delay Request* and will get a response in the form of a *Delay Reply* packet. Based on this two-way handshaking the slave node can obtain all necessary timestamps to determine the time offset from the time master.

Next to time synchronization, traffic scheduling is the second key network feature. In wired TSN, one of the standards used for traffic scheduling is Time-Aware Shaper (TAS) [2]. Before applying TAS, packets from different traffic flows are classified in different hardware queues based on e.g. DSCP values or VLAN tag IDs. TAS then serves each hardware queue for a portion of time during each communication cycle, which repeats periodically. Time slots assigned to hardware queues of a single port of the network device can be shared or dedicated. All timing inside the communication cycle is controlled by the Gate Control List (GCL) that can be programmed at run time from a central controller.

[1]IEEE Time-Sensitive Networking (TSN) task group

III. THE ROAD TO WIFI TSN

Despite the recent advancements introduced by WiFi 7 [12], achieving wire-equivalent dependable and secure wireless communication with time guarantees for usage in time-critical applications remains a major challenge. Moreover, to fully leverage end-to-end deterministic communication, it is important to have a seamless operation between wired and wireless TSN domains. As such, any upcoming wireless TSN features need to be fully compatible with wired TSN features and easy to manage by a unified network controller. As presented in Figure 1, a wired-wireless TSN that makes use of WiFi, is built on top of Ethernet as the communication backbone. This backbone interconnects network management devices such as the Central Network Controller (CNC), wired end-nodes, network switches, and WiFi Access Points (AP), that should support TSN features for the wireless network part. From the network management perspective, CNC is the converging point for managing both wired and wireless TSN domains.

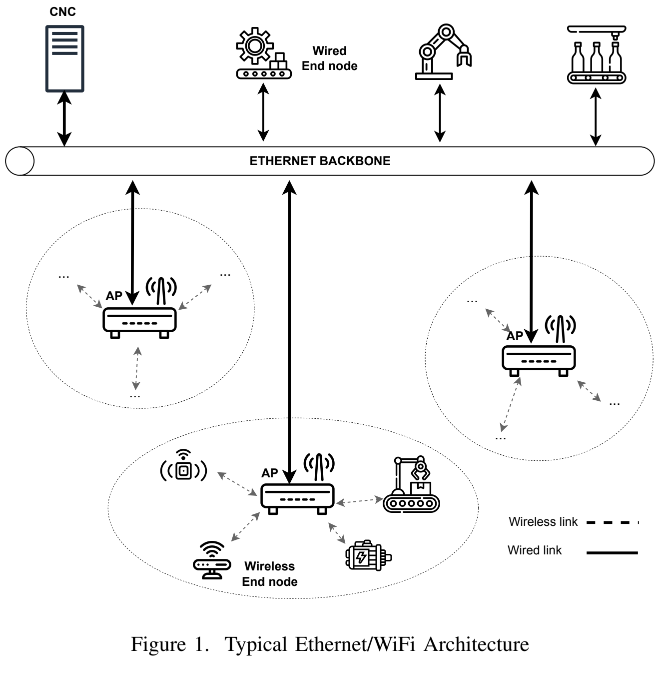

Figure 1. Typical Ethernet/WiFi Architecture

The wireless network domain inherits its beacon-based time synchronization that is currently only used for announcing network presence and for control purposes. Utilization of such mechanism for time synchronization within the wireless domain has been done in the past. However, either the accuracy is not sufficiently high [13] or the synchronization is not achieved end-to-end and is not unified with the wired TSN [14]. In order to support Precision Time Protocol (PTP) over wireless, an accurate timestamping mechanism of PTP-related packets needs to be implemented, which is missing in current WiFi interface cards. To mitigate this, accurate timestamping of PTP packets is needed [15]. Such high accuracy time synchronization can be further utilized for better channel access coordination, a second W-TSN main challenge.

The basic contrasts between wireless and wired communications, such as the changeable capacity of wireless links and the Packet Error Rate (PER) being typically greater in wireless, are some of the issues to be addressed when mapping TSN features to wireless. Another essential factor to consider is the broadcast nature of wireless. On the one hand, it may allow for more devices to be reached with a single broadcast, but it is also more prone to interference. As a result, coordinated access, as well as interference resilience, are critical.

Another aspect to consider is the multiple access method used by WiFi. CSMA/CA tries to maximize the medium usage and lower the amount of collisions by checking whether traffic is present [16]. To move towards WiFi TSN, channel access from different nodes needs to be organized and become deterministic. The IEEE 802.11 standard has introduced schemes to prioritize traffic such as Enhanced Distributed Channel Access (EDCA) in order to support certain Quality of Service (QoS). However, such approaches do not maintain determinism in channel access. In the past, some works such as hMAC [17], have tried to improve channel access determinism by introducing TDMA on top of COTS, by achieving determinism only for downlink communication. In [18] the w-SHARP technology is presented, achieving sub-millisecond real-time communication cycles. Such TSN features are promising, however, in terms of network management and wired-wireless TSN interoperability, there is still a way to go. Other solutions have evaluated TDMA on top of WiFi in a network simulator showing the feasibility of using WiFi for TSN as well [19]. As aforementioned, one of the TSN techniques to achieve reliability is to assign dedicated time slots (TS). Hence, in a fully controlled TSN network, a multiple access method is no longer required for dedicated time slots. However, in a realistic scenario, time resources are limited, and nodes have different requirements and priorities which can lead to dedicated or shared time slot strategies that will be analyzed in Section V.

Control traffic for managing the W-TSN should not only coordinate the node's access to channel resources but also monitor the status of the links. This would inevitably lead to an increase in overhead in the network compared with non-TSN WiFi. Opposite to cellular networks, in WiFi-only a single channel is used for all communications, which might create bottlenecks. For managing the scheduling mechanism in W-TSN an approach would be to incorporate scheduling information in existing WiFi control traffic in the downlink. Further optimization of channel usage can be achieved by enriching the data packets with monitoring information [20].

IV. Wireless TSN based on openwifi

openwifi[2] is the first open-source Linux mac80211 compatible full-stack IEEE802.11/WiFi design based on SDR (Software Defined Radio) [21]. Currently, openwifi includes most of the 802.11a/g/n features together with the freedom to test features such as CCA, retransmissions, contention delays, and more. Currently, WiFi 6 (802.11ax) is under development in openwifi with features like higher modulation and coding schemes, OFDMA, and MU-MIMO. In addition, the platform's flexibility offered ample opportunities to implement and explore new features in view of wireless TSN, including the design of high accuracy hardware timestamping for PTP synchronization and a time-aware gating system.

To have unified end-to-end time synchronization, PTP synchronization over wireless is enabled in openwifi. To achieve the accurate time synchronization in wireless, precise timestamping of PTP-related packets is achieved using the Time Synchronization Functions (TSF) timer [22]. As such, there is no need for a dedicated hardware clock to achieve synchronization. Based on the TSF timer resolution that is 1 μs, the achieved accuracy is in the range of ~ 1 μs [22]. Since the TSF timers of the devices are synchronized with each other via PTP, end device transmission coordination can be controlled using such clock.

openwifi supports four different hardware queues. Each of the queues can support different channel access parameters (contention window (CW) min, CW max, AFIS, and maximum transmission opportunity) as specified by IEEE 802.11e. Such channel access parameters can be dynamically changed at run time for faster access if needed.

In addition to normal priority-based channel access, openwifi supports a gated channel access mechanism for all of the queues. The gate control mechanism is coordinated by the TSF timer. The parameters that are specified by the controller are the communication cycle length, time slot start and time slot end inside the cycle length for each queue as shown in Figure 2. Then the communication cycle is repeated periodically over time. At every TSF increment, the modulo operation of the TSF value and communication cycle length is checked and if the output is 0 then the new communication cycle starts. Then, from the communication cycle start, the time slot start and end are determined. Since the TSF counter accuracy is 1 μs, the smallest time slot that can be applied theoretically is 1 μs. This design is fully compatible with IEEE 802.1Qbv [2].

One additional challenge is to distribute the schedule information from the CNC to the wireless end devices in a compact way, without increasing the control traffic. To achieve this, we determine a set of allowed cycle lengths that can be written in the format $2^n * 512\mu s$, where n is the value that is distributed to the end devices to reduce the amount of bits used. Similarly, the allowed time slot lengths to be used are written in the form $2^n * 128\mu s$ and is a submultiple of defined cycle length. Then the information to be sent to the end device includes the n of the time slot and the time slot ID, defining the position of the

[2]https://github.com/open-sdr/openwifi

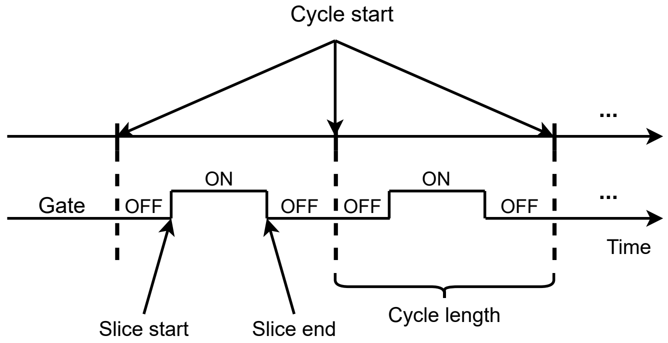

Figure 2. W-TSN scheduling system

Table II
SCHEDULING NUMEROLOGY

Cycle n / TS n	0	1	2	3	4	5	6
0	4	8	16	32	64	128	256
1	2	4	8	16	32	64	128
2	1	2	4	8	16	32	64
3	0	1	2	4	8	16	32
4	0	0	1	2	4	8	16
5	0	0	0	1	2	4	8
6	0	0	0	0	1	2	4
7	0	0	0	0	0	1	2

time slot inside the cycle. The numerology of such scheduling information is given in Table II, where cycle and time slot n are given as well as the number of time slots per cycle.

V. RESULTS

Using the wireless TSN building blocks presented in Section IV, a preliminary assessment of scheduling strategies for uplink traffic is performed.

As demonstrated by different works, TSN scheduling to guarantee bounded latency and high reliability is a complex problem [23], [24]. Considering every node has its own requirements, with diverging traffic flow priorities, plus a constantly changing environment and limited time resources, it is important to devise strategies to optimize time slot usage.

As not all traffic flows are high priority, and TSN aims for coexistence between time-sensitive and best-effort traffic, time slot sharing becomes a clear option to optimize resource usage. However, not only the traffic priority plays a role in the sharing/not-sharing decision. A wireless end node has variables such as the link data rate, application packet size, traffic types, assigned time slot size, and others that should also be taken into consideration.

In case of dedicated time slots, there is only a single node transmitting in the time slot and contention mechanisms like CW, NAV, $DIFS$, or $EIFS$ are not required anymore. Hence, equation 1 presents the necessary time for transmitting a 118 bytes packet at 65 Mbps link rate using IEEE 802.11n:

$$\begin{aligned} Total_D &= T_{DATA} + T_{SIFS} + T_{ACK} \\ Total_D &= 14.52 + 13 + 93.33 \\ Total_D &= 120.85\mu s \end{aligned} \tag{1}$$

As shown in Equation 1, T_{ACK} presents the largest delay. This is because the ACK, as all control packets, is transmitted at the lowest data rate, in this case 6 Mbps.

Alternatively, a shared time slot requires the aforementioned contention mechanisms. Besides this, the variable nature of the channel or hidden node problem might introduce collisions, which lead to bigger delays due to re-transmissions and back-off.

$$\begin{aligned} Total_S &= T_{DIFS} + T_{DATA} + T_{SIFS} + T_{ACK} + CW \\ Total_S &= 28\mu s + 14.52\mu s + 13\mu s + 93.33\mu s + CW \\ Total_S &= 148.85\mu s + CW \end{aligned} \tag{2}$$

Equation 2 shows the delay of a single uplink packet transmission in a shared time slot. Again, IEEE802.11n is used with a packet of 118 bytes. In comparison with Equation 1, $DIFS$ and CW are added. In this Equation, CW represents back-off. From $Total_D$ and $Total_S$ we already notice the theoretical advantage of a dedicated time slot to fit more packets per cycle. Having introduce both scheduling options, the measurements comparing these two scenarios are presented in the following subsections.

A. Setup Description

As introduced in Section IV, the openwifi IEEE802.11/WiFi baseband chip/FPGA design, along with the TSN extensions, is used on top of the ADRV9361-Z7035 Software Defined Radio (SDR), which combines the Analog Devices AD9361 integrated RF Agile Transceiver and the Xilinx Z7035 Zynq®-7000. An Ethernet interface is added using the carrier card ADRV1CRR-BOB/FMC [21].

The infrastructure mode wireless network, as shown in Figure 3, consists of three nodes: one access point (AP) and two stations, all of which are connected to a PC acting as the Central Network Controller (CNC) for the W-TSN.

For the wireless environment, a WiFi channel 36 with 20MHz bandwidth is used. No other nodes are using this channel or any overlapping channels during the experiments. Hence, no external interference is considered during the experiment.

B. Shared vs. Dedicated Time Slot Measurements

During the measurements all the network nodes are synchronized using PTP over WiFi [22]. The AP is set to be the PTP master node, while the stations are PTP slave nodes. Furthermore, as shown in Figure 4, PTP traffic, UL and DL, is confined to a different time slot inside the communication cycle, not to interfere with the data traffic.

The selected cycle length is $65535\mu s$, and the time slot (TS) size ranges from $128\mu s$ to $1024\mu s$. In line with the theoretical calculations in Equations 1 and 2, a fixed data rate of 65 Mbps

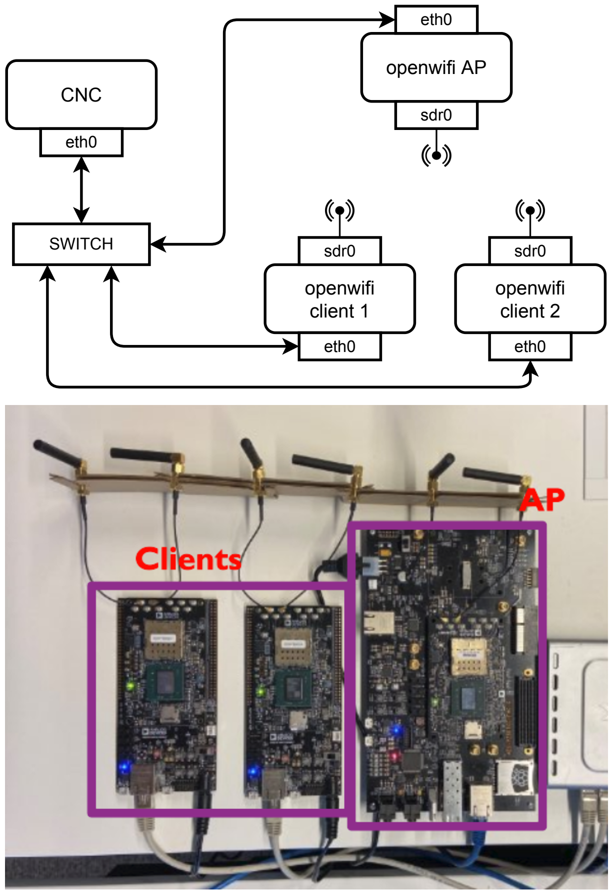

Figure 3. Setup Architecture.

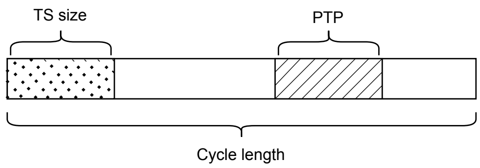

Figure 4. Measurements schedule

is used with IEEE802.11n. The injected traffic emulates one of the typical time-sensitive applications, namely machine control traffic. A total of one thousand UL packets are periodically generated every 1 ms for every experiment.

In the non-shared measurements, all contention mechanisms are disabled, hence once a TS is available and the client has a packet queued, it will transmit immediately. Figure 5 shows the results of mode values for both measurements.

In Figure 5, the purple bar represents the number of packets that arrived at the AP from a unique client not sharing its TS. As expected, the longest the TS, the higher the number of arrivals. The blue and orange bars represent the number of arrived packets at the AP when two clients, (C_1 and C_2), share the TS. Finally, the yellow bar represents the addition of both clients' arrivals in the shared TS experiment.

It is clear from Figure 5 how the contention produced by

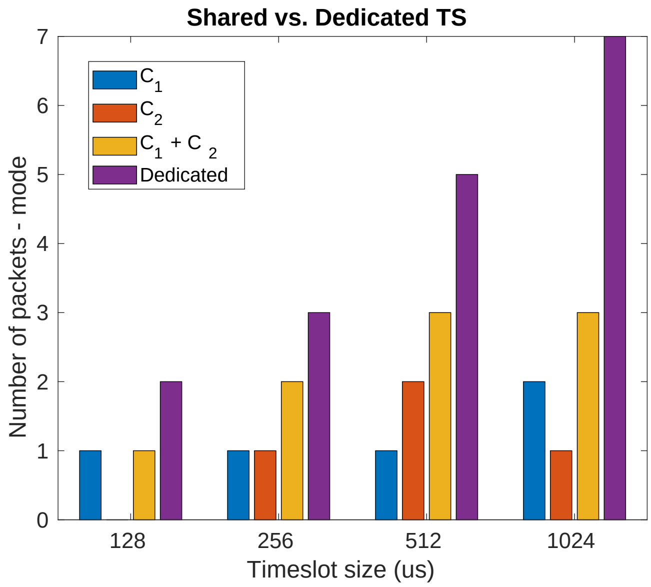

Figure 5. Received packets per time slot size

both clients sharing the TS, decreases the number of packets transmitted and thus received by the AP compared with the non-shared case. It is also noticeable how this difference becomes bigger when the TS size increases.

Complementary to this, it is also clear that Equations 1 and 2 certainly reflect the behavior captured with the experiments.

Table III
CYCLE THROUGHPUT

	Shared (bps)			**Dedicated (bps)**
TS size (us)	C_1	C_2	$C_1 + C_2$	
128	14.404	0	14.404	28.808
256	14.404	14.404	28.808	43.212
512	14.404	28.808	43.212	72.02
1024	28.808	14.404	43.212	100.828

Table III presents a comparison of cycle throughput (CT) for both experiments using the number of received packets at the AP - mode. CT was calculated considering UL packet sizes of 118 bytes and a cycle length of 65535μs. As this is almost completely dependent on the number of packets, the tendency is the same as in Figure 5.

After examining the numbers of Table III, it might be concluded that the CT is rather low compared with current wireless communication systems. However, the experiment's schedule represents an extreme case. In a normal scenario, higher data rates, bandwidths, and TS would be used, as well as smaller cycles and more TS per cycle, increasing the achievable throughput. In this experiment the assigned time percentage ranged from 0.07% for TS 128 μs case to 1,56 % for the TS 1024 μs case. Such small percentage of air time clearly show that the achieved CT depends on the assigned schedule.

VI. Conclusion

In this paper, we presented some of the main challenges that need to be overcome to achieve a seamless Ethernet/WiFi TSN network. Despite the industry is still skeptical that the same performance level in synchronization, latency, security, and reliability can be achieved over wireless, TSN features on top of WiFi are bringing wireless closer to this deterministic vision.

Still, additional research effort is required in features such as scheduling in a wireless environment. The variable nature of the channel together with the limited amount of resources requires adaptable mechanisms that are not only able to allocate time-sensitive but also best-effort traffic. Then, it is important to consider instruments such as time slot sharing for lower priority streams.

Using our TSN features built on top of openwifi, a small-scale W-TSN testbed was built and measurements were conducted to compare shared and dedicated time slots. Together with a complementary theoretical analysis, it was demonstrated how contention produced by CSMA/CA in a shared time slot, directly impacts the received throughput.

Regarding future work, a study considering variables such as data rate, different packet sizes, generation time effects, higher bandwidths, and bigger time slot sizes with optimization and machine learning techniques will be performed.

Acknowledgment

This research was funded by the Flemish FWO SBO S003921N VERI-END.com (Verifiable and elastic end-to-end communication infrastructures for private professional environments) project and by the Flemish Government under the "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen" program.

References

[1] "802.1AS-2020 - IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications," Tech. Rep., 2020.

[2] "IEEE 802.1Qbv-2015 - IEEE Standard for Local and Metropolitan Area Networks - Enhancements for scheduled traffic : bridges and bridged networks," p. 57, 2016.

[3] "IEEE Std 802.1Qbu-2016 (Amendment to IEEE Std 802.1Q-2014) : IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks – Amendment 26: Frame Preemption." 2016.

[4] "IEEE Std 802.1Qci-2017 - IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks–Amendment 28: Per-Stream Filtering and Policing," pp. 1–65, 2017. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=8064219

[5] "802.1CB-2017 - IEEE Standard for Local and metropolitan area networks–Frame Replication and Elimination for Reliability," 2017.

[6] "IEEE P802.1Qcc - IEEE Draft Standard for Local and metropolitan area networks–Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements," pp. 1–236, 2018.

[7] M. Nixon, "A Comparison of WirelessHART and ISA100.11a," 2012.

[8] M. Chwalisz and A. Wolisz, "Towards efficient coexistence of IEEE 802.15.4e TSCH and IEEE 802.11," *IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018*, pp. 1–7, 7 2018.

[9] F. Grijpink, A. Ménard, H. Sigurdsson, and N. Vucevic, "The road to 5G: The inevitable growth of infrastructure cost," *McKinsey&Company*, p. 8, 2018.

[10] 5G PPP Architecture Working Group, "View on 5G Architecture," 2019.

[11] "802.1Qcr-2020 - IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks - Amendment 34 Asynchronous Traffic Shaping." 2020.

[12] 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, p. 4954, 7 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/15/4954

[13] J. Haxhibeqiri, X. Jiao, M. Aslam, I. Moerman, and J. Hoebeke, "Enabling TSN over IEEE 802.11: Low-overhead Time Synchronization for Wi-Fi Clients," in *Proceedings of the IEEE International Conference on Industrial Technology*, vol. 2021-March. Institute of Electrical and Electronics Engineers Inc., 3 2021, pp. 1068–1073.

[14] A. Mahmood, R. Exel, and T. Sauter, "Performance of IEEE 802.11's Timing Advertisement Against SyncTSF for Wireless Clock Synchronization," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 370–379, 2 2017.

[15] I. V. Óscar Seijo, Jesús A. López-Fernández, and Manuel Vélez, "IEEE 1588 Clock Synchronization Performance over Time-Varying Wireless Channels," 2018.

[16] D. J. G. Akin, *CWAP: Certified Wireless Analysis Professional ™ Official Study Guide*, 1st ed., McGraw-Hill/Osborne, Ed. Brandon A. Nordin, 2004.

[17] S. Zehl, A. Zubow, and A. Wolisz, "hMAC: Enabling Hybrid TDMA/CSMA on IEEE 802.11 Hardware," no. November, 2016. [Online]. Available: http://arxiv.org/abs/1611.05376

[18] O. Seijo, J. A. Lopez-Fernandez, and I. Val, "W-SHARP: Implementation of a High-Performance Wireless Time-Sensitive Network for Low Latency and Ultra-low Cycle Time Industrial Applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3651–3662, 2021.

[19] O. Seijo, J. A. Lopez-Fernandez, H. P. Bernhard, and I. Val, "Enhanced Timestamping Method for Subnanosecond Time Synchronization in IEEE 802.11 over WLAN Standard Conditions," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5792–5805, 9 2020.

[20] J. Haxhibeqiri, X. Jiao, E. Municio, J. M. Marquez-Barja, I. Moerman, and J. Hoebeke, "Bringing Time-Sensitive Networking to Wireless Professional Private Networks: Filling Gaps and Bridging the Innovation," *Wireless Personal Communications*, vol. 121, no. 2, pp. 1255–1271, 2021.

[21] X. Jiao, W. Liu, M. Mehari, M. Aslam, and I. Moerman, "Openwifi: A free and open-source IEEE802.11 SDR implementation on SoC," *IEEE Vehicular Technology Conference*, vol. 2020-May, 5 2020.

[22] M. Aslam, W. Liu, X. Jiao, J. Haxhibeqiri, G. Miranda, J. Hoebeke, J. Marquez-Barja, and I. Moerman, "Hardware Efficient Clock Synchronization Across Wi-Fi and Ethernet-Based Network Using PTP," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 3808–3819, 6 2022.

[23] V. Gavrilut and P. Pop, "Scheduling in time sensitive networks (TSN) for mixed-criticality industrial applications," *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*, vol. 2018-June, pp. 1–4, 7 2018.

[24] A. M. Kentis, M. S. Berger, and J. Soler, "Effects of port congestion in the gate control list scheduling of time sensitive networks," *Proceedings of the 2017 8th International Conference on the Network of the Future, NOF 2017*, vol. 2018-Janua, pp. 138–140, 7 2017.