Operational Impacts of IEEE 802.1Qbv Scheduling on a Collaborative Robotic Scenario

Richard Candell*, Karl Montgomery*, Mohamed Kashef (Hany)*, Susruth Sudhakaran†, Justin Albrecht*, and Dave Cavalcanti †
*Smart Connected Systems Division, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA
†Intel Labs, Intel Corporation Hillsboro, OR, USA

***Abstract*—Time-sensitive networking (TSN) is an emerging topic for the advancement of wireless networking for industrial applications. TSN, as defined under the umbrella of IEEE 802.1 working group standards, addresses issues related to providing deterministic communications over IEEE 802-based Local Area Networks (LANs). TSN was originally designed to support real-time audio/video applications over Ethernet providing better reliability and lower, more deterministic latency with traffic shaping capabilities. TSN has since expanded its scope and applicability to other applications such as those in industrial environments and automotive applications. Industrial examples include machine-machine communications for robot control, end-effector actuation, real-time sensing, and safety integrated systems. Applications utilizing a wireless local area network (WLAN) can also benefit from scheduling and traffic shaping as defined in the 802.1Qbv standard; however, factors such as clock stability, synchronization, resource requirements and protocol options come into play when selecting a schedule to support multiple application types on the same network. In this article, we present a scenario for a collaborative robot heavy lift operation, in which, two robots communicate over an IEEE 802.11 WLAN with TSN capabilities to lift a rigid body in three dimensions. Scheduling is performed using 802.1Qbv over WLAN with the robot operating system (ROS) used as the software middleware utilizing the transport control protocol (TCP). As a part of the research, we describe our process for schedule selection to accommodate the time-sensitive traffic of the robotic scenario while allowing an industrial internet of things (IIoT) high data rate traffic to coexist. We then provide an analysis of the impacts of TSN schedule selection on the operational performance of the collaborative robot application.**

***Index Terms*—Wireless, TSN, factory communications, IEEE 802.11, IEEE 802.1Qbv, WLAN**

I. Introduction

A. Industrial wireless need

The vision of the new industrial revolution includes a much wider deployment of digital Information Technology (IT) applications such as Artificial Intelligence (AI) and Internet of Things (IoT) [1]. However, these IT applications are required to share various resources with the operational technology (OT) equipment. As a result, the integration of IT and OT domains needs strictly time-synchronized and deterministic low latency communications [2]. Industrial wireless promises to achieve the needed flexibility, easy reconfigurability, and mobility of the future industrial systems. Moreover, wireless time sensitive networking (TSN) [3], [4] has emerged to complement these benefits by achieving the required time synchronization and timeliness in wireless networks.

B. Testbed Description

For this work, we utilize the National Institute of Standards and Technology (NIST) industrial wireless testbed to perform experiments and measurements. The testbed consists of two industrial-grade robots, which are two Universal Robot UR3s, that oppose each other. One is the leader robot, and the other follows the leader's path using a velocity-based controller. The network was configured as completely wired (Ethernet) or a hybrid of wired and wireless using both Ethernet and Wi-Fi connections. The specifics of the Dual-Lift robotic use case are discussed further in Section II. All network data is captured using tap devices. The data collection machine and other relevant nodes are globally synchronized to a Grand Leader (GL) clock to enable precision latency measurements. We also take time-synchronized robot position measurements to compute the Cartesian position error between the leader and follower end-effectors using the robot controller's data stream through the real-time data exchange (RTDE) interface. In this way, we are able to compare the physical performance of the use case to the network performance using a synchronized and shared time. Emulated industrial internet of things (IIoT) traffic is conducted through the same network to study the impact of co-existing high-density IIoT traffic while facilitating the optimization of the schedule to accommodate more IIoT devices while also maintaining the performance targets of the collaborative robot operation.

C. Wireless TSN overview

TSN refers to a group of networking-related protocols and standards developed by the IEEE 802.1 working group in order to provide reliable, and time bounded (deterministic) delivery of data over 802-LAN (Local Area Network) technologies. As the 802.11 technology is also a LAN technology, the TSN concepts can be mapped from Ethernet to Wi-Fi without major architectural changes or translation gateways. However, the use of unlicensed frequencies, random medium access (listen before talk) in 802.11 imposes some challenges to accomplishing this mapping. In this paper, we evaluate an implementation of the main TSN capability that has been enabled across wired and wireless links, which is time-aware scheduling based on the IEEE 802.1Qbv standard. The purpose of the 802.1Qbv standard, therefore, is to define a transmission schedule that segments different classes of traffic based on their sensitivity to delay and jitter by reserving specific time windows for time-sensitive traffic flows. An overview of similar mapping and implementation efforts over the 5G technology are described in [10].

D. Contributions in this paper

To summarize our contributions in this paper, a) we have developed an industrial collaborative testbed where industrial operational metrics can be measured and high-data-rate IIoT traffic can coexist with the time-critical testbed traffic, b) we demonstrate an approach for 802.1Qbv schedule selection to maintain the required testbed operational performance while allowing the IIoT traffic to be transferred over the same wireless network, and c) we demonstrate the need for TSN schedule protection to be propagated through the entire protocol stack to achieve the required response in specific scenarios. In the results, we study the impact of the protected window size for the leader-follower stream on the overall testbed performance from both the wireless network and operational viewpoints. Moreover, we analyze the impact of the schedule selection on the coexistence with other IIoT best-effort traffic.

E. Previous work

In previous work [6], we have demonstrated the improvement in latency in an industrial robotic machine tending use case when interfering traffic is introduced in the same network, using a time-based schedule with globally time-synchronized nodes. In that work, we demonstrated the ability to use TSN over Wi-Fi to achieve a bounded latency of 5 ms with a probability of 99%. However, the previous testbed scenario did not employ high-data-rate traffic streams emulating a high density of competing IIoT devices. In other previous works, such as [8], we showed how an industrial robotic use case wireless network is affected by interfering traffic without wireless TSN techniques. However, our previous works utilized a pick and place supervisory control-based application [9], whereas this paper uses a new use case, which utilizes a closed-loop regulatory control application. This type of application has stricter latency requirements, as mentioned in NIST's wireless user requirements for the factory workcell [11]. The follower robot receives a real-time position information stream, sent by the leader robot, used to control its movement. Currently, wireless TSN in the context of robotics and industrial automation is a novel field, as the technology has not yet been adopted on a large scale.

II. Dual-Lift Robotic Use Case

The dual-lift robotic use case performs coordinated movements with a leader robot and a follower robot to jointly lift and move an object. See Fig. 1 for a picture of the testbed. The object lifted is a custom metrology bar with an integrated force-torque sensor (ATI mini45), which is held by

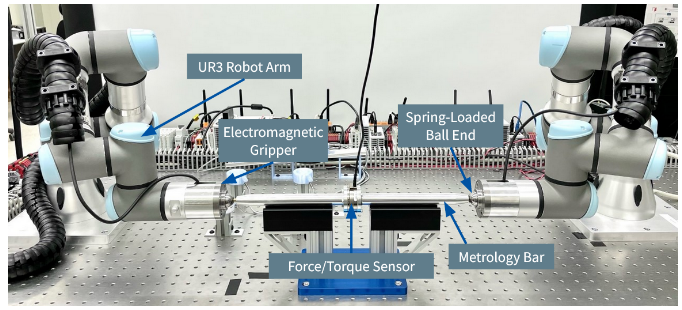

Fig. 1. Dual Robotic Use case performing a circle pattern

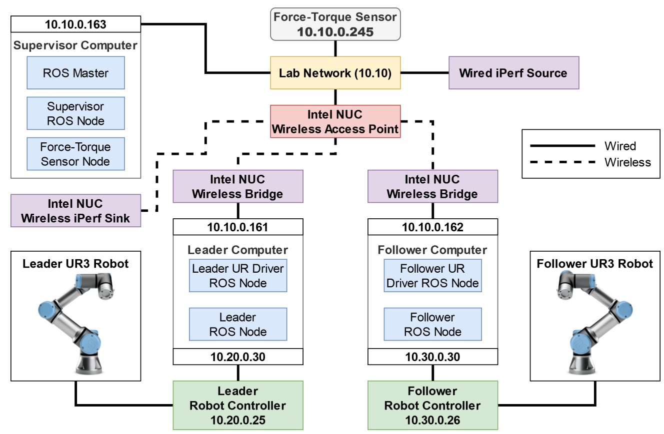

Fig. 2. Network diagram of the Dual-Lift use case

the robots utilizing custom electromagnetic grippers with a ball and socket connection. The bar has spring-loaded ends, such that the robots will not drop the bar in extreme cases of leader-follower position error. The robots are two Universal Robots, UR3s, and are directly controlled by Robot Operating System (ROS) nodes, implemented on SuperLogics microbox-PCs, running Ubuntu 18.04 and ROS Melodic. ROS is a communication middleware that was used to control and coordinate the motion of the robots. ROS is a distributed system, in which various nodes can be instantiated to communicate over a network through topics, services, and actions. More information about ROS can be found here [12]. For brevity, the inner workings of the program will not be discussed, but we will discuss the relevant data flows to the leader and follower ROS nodes.

Fig. 2 shows the network diagram of the use case, which includes the connection information for the relevant devices in the testbed. The ROS nodes are listed under each relevant computer. The UR driver nodes for the leader and follower are what communicate with the leader and follower robot controllers to move the robots as well as actuating the grippers. Additionally, there is a number of Next Unit of Computing (NUC) machines. The Ethernet-wireless bridges are implemented using NUCs, which serve as the wireless stations (STAs), and one NUC is configured as the wireless access point (AP). They have 2x2 MIMO wireless capability with IEEE 802.11ac Wi-Fi cards. The AP was configured to use a

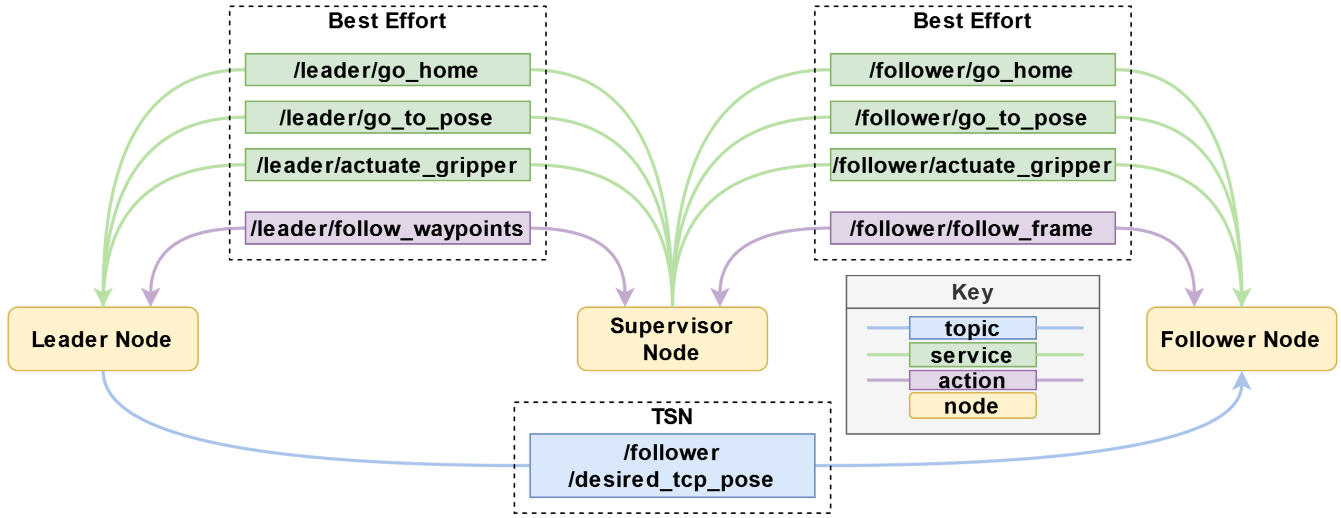

Fig. 3. Diagram of ROS-based software implementation of a leader-follower control system. ROS utilizes a decentralized TCP-based publish-subscribe message topic with independent clock-triggered message transmission.

20 MHz channel and operated at Modulation Coding Scheme (MCS) 15, with a theoretical max speed of 130 Mbps. Also, all of the NUCs for the wireless bridges and wireless AP are synchronized over Ethernet using the IEEE 1588 Precision Time Protocol (PTP) [5], which allows for each device to keep the TSN schedule's timing, within $<$1 us of error from the GL clock with 99% confidence. In previous work, we have shown that we can synchronize wirelessly with IEEE 802.1AS within $<$100 us error from the GL in [6], however, we did not want to introduce another variable that would impact our results, as we are studying the schedule impact on the network performance.

The data streams, which are collected using Ethernet tap devices called SharkTaps, are routed to the collection machine to capture the network packets. These tap points are not shown in the figure. This allows us to capture the times on a collection machine that is globally synchronized to the GL clock. Also, a wired traffic source and wireless traffic sink are shown in the figure, which use iPerf [7] to create a wireless traffic stream with 1000 Byte (B) length packets over the User Datagram Protocol (UDP) wirelessly from the AP to the sink STA.

The ROS diagram of various nodes and data streams is presented in Fig. 3. The follower's control algorithm utilizes a velocity-based controller, with real-time data from the leader, using the desired_tcp_pose topic streamed at 125 Hz. This topic contains the position and orientation information of where the follower should move next, calculated from the bar's length. This topic is opened at a pre-defined port number, and the AP uses this port number to protect this traffic in its corresponding window. Also, this data stream directly impacts the follower movement, as any significant latency disturbances or loss in this stream will cause the follower to jerk or even stop briefly. Since this stream is a ROS topic, the protocol used is Transmission Control Protocol (TCP). This application is latency sensitive by design, as we are able to see and measure small latency disturbances in real-time with the leader-follower error. TCP and TSN will be discussed later.

III. 802.1Qbv Scheduling Approach

A. Schedule Definition

As discussed in the previous section, there are multiple streams in the use case being considered. The most critical stream among them is the communication between the leader node and the follower node, whereby the leader conveys the position that the follower should move to. It is important to ensure that this communication happens in a deterministic fashion and the presence of the other traffic streams does not affect the timely delivery of position data to the follower. This is accomplished by defining a traffic schedule and synchronizing this schedule in time across all the participating wireless nodes. In general, two classes of traffic are being defined - Time Critical (TC) traffic and Best Effort (BE) traffic.

In this use case, we let this most critical data stream to be the only TC stream in the schedule. All other traffic streams except the leader-follower communication have been assigned to the BE class. Fig. 4 shows a high-level definition of such time-aware schedule that is implemented at each node sharing the medium. The schedule is characterized by a Cycle Time (T), which is the period at which the schedule repeats. Each period of the schedule is divided into time slots for TC traffic and BE traffic. During a particular time slot, only traffic belonging to that class is allowed to be transmitted or pass through and all other traffic streams are queued. This gating rule is accomplished by a set of gates that open and close based on absolute time. In this use case, the cycle time is 8 ms because this is the periodicity of the position updates from the leader to the follower, which is the TC traffic in this case. The slot times have to be derived by carefully observing the traffic profile of the stream and its time footprint in the wireless medium. The slot times also have to strike balance between meeting the minimum timing requirement of the TC traffic for optimum performance and the requirements of the best effort traffic such that, although the TC traffic is the higher priority traffic, the BE traffic requirements should not be completely ignored. In the figure, we also see a slot where no traffic is let through. This slot is called the guard interval and is required to make sure that the traffic in the BE slot does not encroach into the next time period in the schedule. As Wi-Fi uses random medium access, it is important to ensure that this schedule is synchronized to and implemented at each WiFi node. It is also important to have a common notion of time across all of these nodes so that the slot times defined and implemented by the schedule are synchronized across all the nodes.

B. Wireless TSN Schedule Calculations

To determine the protected window and guard band slot sizes to be used in the schedule, we need to understand the transmission characteristics of the streams belonging to each slot. If we look at our TC traffic, the leader-follower communications, we see that it is a TCP stream where a position update packet is sent at 125 Hz as a TCP packet. For the TC slot for this traffic, we have to make sure that the worst-case duration of this exchange, which consists of a TCP packet and its acknowledgment (ACK) packet, is within the slot duration. If this condition is not met, then this will affect the rate of position update as the TCP algorithm will scale back and eventually this will impact the performance at the application level.

We measured the transmission time of the packet exchange using a real-time spectrum analyzer (RTSA) to derive the maximum duration of the leader-follower per packet transmission. This duration was observed to be at 88 us at MCS 15, but assuming a worst-case MCS index of 13, our base protected window size was set at 110 us. Normal operation of the Wi-Fi has been observed to be consistently at the maximum MCS index of 15, due to the high signal-to-noise ratio in the lab environment. With this base number of 110 us, we conducted multiple experiments by letting the protected window size to be a multiple of 110 us with various scaling factors. The goal of this experimental procedure is to find the scaling factor that achieves the optimum performance of the use case as measured by the error vector magnitude (EVM) measurement, which is the error between the expected position of the follower and its actual position.

The protected window multipliers used for our experiments are 5x, 10x, 20x, 30x, and 40x. At a smaller multiplier than 5x, the use case would run into errors and at a 40x multiplier, the desired performance is reached. Our guard band for the schedule was set at 334 us (assuming a worst case of MCS 13), based on the observed maximum packet length of 273 us from the BE at MCS 15. The minimum TSN schedule multiplier that performs under the selected EVM was then evaluated at different interfering traffic levels to observe its performance under harsh network conditions. The selection of the protected window multiplier is further discussed in Section V by using the results presented.

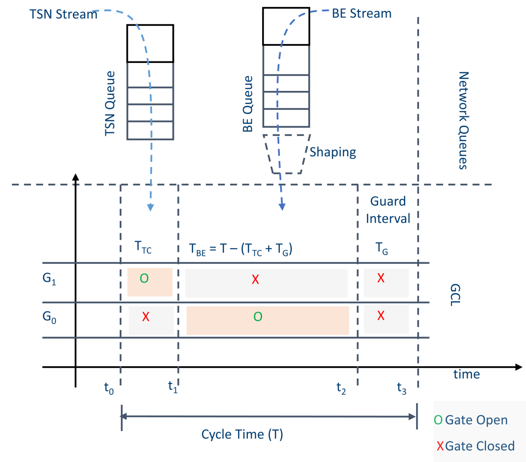

Fig. 4. Time Aware Traffic Schedule using IEEE 802.1Qbv

IV. Measurement and Experimental Methodology

This section describes the measurement methodology. The results presented in the paper consist of physical and network performance data. The ground truth error between the leader and follower represents the physical performance, whereas the latency analysis represents the network side. The measurement machine, to which the SharkTaps are routed, collects the network packets along with RTDE, [13], data at 125 Hz from the robots, to calculate the ground truth error between the leader and follower. For the measurement machine, the clocks on multiple 4-port gigabit Ethernet PCI cards, based on Intel i210, are time synchronized to the GL clock using PTP time synchronization for <1 us error with 99% confidence. The ground truth error is also accurate in time synchronization, as the same collection machine is used for both RTDE data streams from the leader and follower robot controllers.

Each measurement was taken for the duration of 40 revolutions of a circle, taking approximately two and a half minutes to finish. A circle is chosen as the error between the leader and follower under perfect conditions should be a constant value, due to the constant radial acceleration. The experiments are differentiated based on their wired or wireless configuration, the level of competing IIoT traffic, and the TSN schedule configuration. The IIoT traffic is generated through iPerf, emulating sensor data. We chose this stream to have 1000 B length packets and to stream at 16, 32, 48, 64, and 80 Mbps levels, as the higher competing traffic impacts the performance of the use case significantly without TSN enabled. These runs were performed using the standard Wi-Fi carrier-sense multiple access (CSMA) without TSN features for later comparison with the TSN capabilities at similar traffic levels. Then, the wireless TSN schedule was enabled, with various protected window lengths at 5x, 10x, 20x, 30x, and 40x multipliers. Lastly, we determined the best protected window multiplier, and used the same interference levels from CSMA (no TSN) with competing traffic to show the performance of a single TSN schedule with varying BE traffic.

Two types of key performance indicators (KPIs) are considered in this work, namely, the physical and network performance metrics. The physical error is an example of the physical KPIs and is presented by measuring the cumulative density function (CDF) of the Cartesian error magnitude between the leader and follower robots. The Cartesian error magnitude is defined as the physical distance between the follower position, at any moment, to the required position in mm. Due to the various stochastic variations in the experiment and the environment, we use the CDF as a statistical measure of how frequently the error takes various values and how deterministic the error is. On the other side, as an example of the network KPIs, we present the CDF of the Inter Arrival Rate (IAR) of the TCP packets of the leader-follower traffic stream. The IAR of the TCP packets is the time difference between every two consecutive packets of the stream at which the data is ready at the TCP layer and this value is affected by both application data generation and the underlying communication data handling. Hence, it represents the impact of the TSN schedule on the TC data stream. Similarly, we deploy the CDF to capture the statistical change of this KPI.

V. Results

The Dual-Lift use case experiments were run with the various network configurations and interfering traffic explained in Section IV. We also include a wired baseline for reference, as this is what is traditionally used for communication with industrial robots. The dataset of the collected data during this experimental study is available online [14]. The first two CDF plots of the RMS Cartesian error of the leader-follower are shown to determine the best multiplier for the schedule, targeting an EVM $<$15.7 mm at 95% probability. This EVM threshold was set arbitrarily, as our use case can allow for higher errors without breaking the held object, however one can imagine that real applications are not as forgiving as ours, since our metrology bar can lengthen if needed.

We can see from Fig. 5 that 48 Mbps CSMA is above the EVM threshold, thus this is our breaking point level, which we center our various interfering traffic levels around. Also note how the performance rapidly degrades with higher interference levels without TSN, such as with 64 Mbps and 80 Mbps.

Next, in Fig. 6, we can see that when the TSN capability is enabled, the required protected window size is enabled with a least a multiplier of 30x, as that is the level that is below the 15.7 mm EVM threshold. Such window size (3 msec) accounts for the channel access overhead and any other software stack overheads. In Fig. 7, we show the IAR of the TCP packets at various schedule lengths. The 5x multiplier can clearly be seen to be inducing extra latency between packets, which means the follower is failing to receive the packets at 125 Hz, or every 8 ms, effectively reducing the update rate of the data stream. Higher window sizes of 20-40x perform better in this aspect and deliver the majority of the packets at 8 ms.

In Fig. 8, using the minimum best protected window multiplier of 30x, we ran the levels of interference (16 - 80 Mbps) centered around the EVM breaking point of 48 Mbps and measured the RMS Cartesian error. We can see that the performance now with TSN starts to worsen at 64 Mbps, but this result is still improved compared with CSMA without TSN. Also, we observed much less jerkiness from the robots, as the latency is much more deterministic.

The reason to why 64 Mbps and 80 Mbps degrade performance is because these levels reduce the protected window size in our application. Since the AP buffers the iPerf traffic, if the traffic level is higher than 52 Mbps, we observed the protected window shrunk to accommodate the buffered traffic.

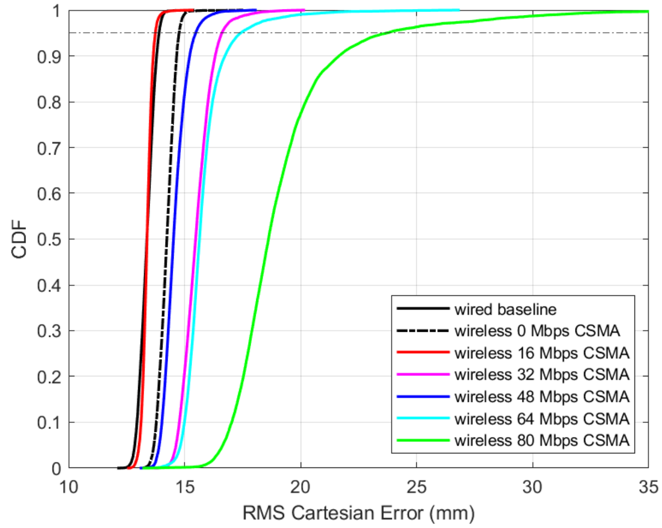

Fig. 5. CDFs of RMS error of Cartesian error by Leader and Follower under varying levels of BE iperf traffic using CSMA indicated by "xx Mbps" in the legend. Here we determine the aggregate bit rate at which the leader-follower violates the EVM threshold.

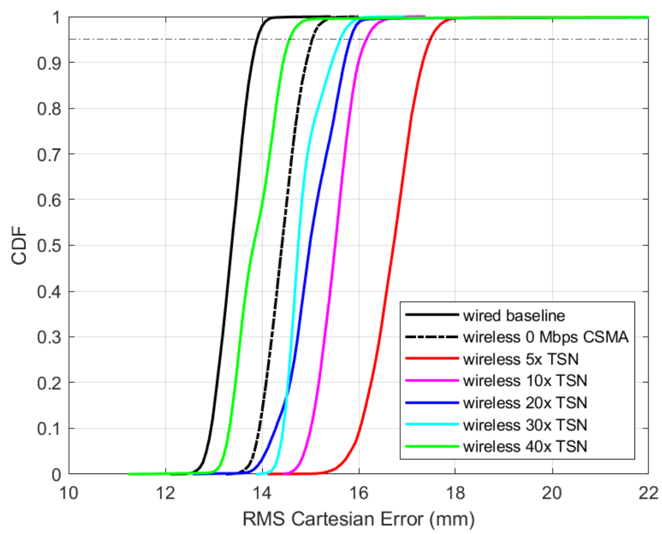

Fig. 6. CDFs of RMS Error of Instantaneous Leader-Follower end-effector Cartesian positions for varying robot traffic protected window multipliers. Generally, RMS error is reduced as the round-trip delay is reduced by accommodating the TCP ACK within the same window and packet inter-arrival jitter is more tightly controlled.

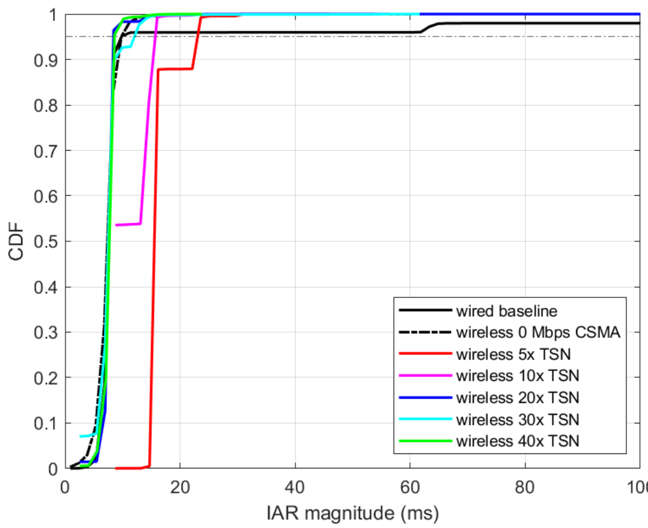

Fig. 7. CDF of Inter Arrival Rate (IAR) of data packets in milliseconds, captured at the follower robot's network tap. Protected windows of 5-40x were used.

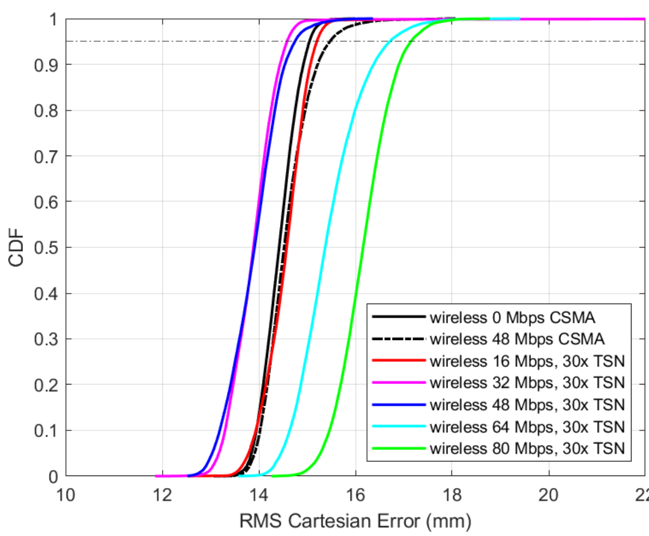

Fig. 8. CDFs of RMS Error of Instantaneous Leader-Follower Cartesian position, keeping the protected window fixed at 30x and varying the level of background traffic. Cartesian error is maintained below the 15.7 mm threshold for background traffic levels below 48 Mbps. Higher levels of traffic begin to intrude on the protected window indicating a need for bandwidth control and Qbv synchronization between protocol layers.

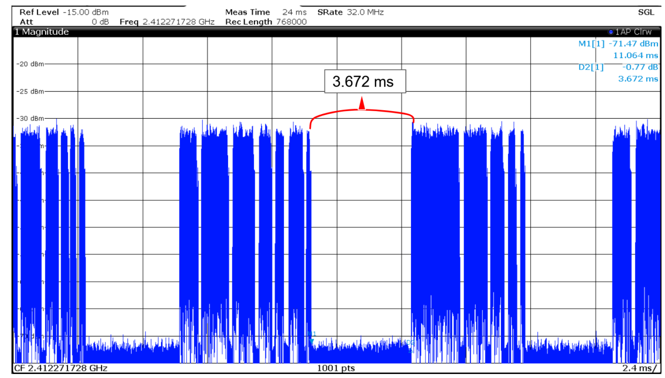

Fig. 9. RTSA time domain capture with 30x protected window multiplier and 48 Mbps of background traffic. Here it is shown that the Qbv protected window traffic is sufficiently insulated from the segregated background traffic.

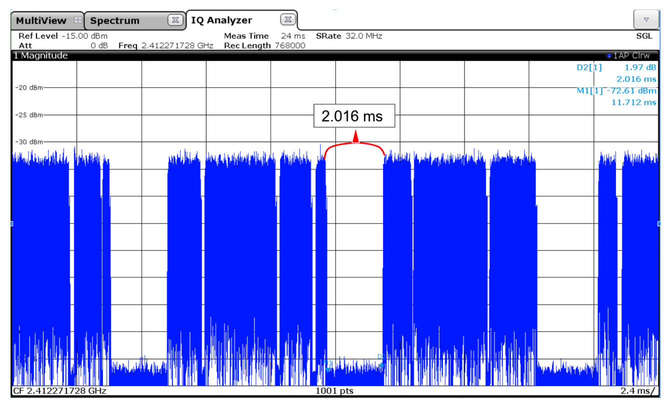

Fig. 10. RTSA time domain capture with 30x protected window multiplier and 80 Mbps of background traffic. As shown, Qbv solely applied at the network layer cannot prevent the intrusion of background traffic into the protected window indicating the need for Qbv synchronization between the network and media access control layers.

Hence, instead of the BE traffic being lost at higher than 52 Mbps of interfering traffic, the protected window shrinks from the measured size of 3.672 ms (configured as a 3.3 ms protected window + 0.334 ms guard band), as seen in Fig. 9 to 2.016 ms in Fig. 10. We also observed that at the maximum speed of our wireless link using iPerf, which was approximately 104 Mbps, the protected window shrinks down almost completely. For this reason, we suggest that additional 802.11 features, such as triggered-based scheduling defined in 802.11ax and further enhancements in the next generations of the Wi-Fi protocol to ensure that TSN schedules can be guaranteed at the 802.11 MAC layer, and not only at the network layer as used in this work. This way the protected window can be guaranteed to be protected by the AP and STAs in the 802.11 networks, increasing dependability and efficiency.

VI. Conclusion

In this work, we applied wireless TSN capabilities enabled in an 802.11 network to a new collaborative robotic use case in an industrial wireless network testbed. We studied the impact of various levels of resource allocation to protect time-critical streams in a shared network with other IIoT best-effort traffic. The results illustrate that the 802.1Qbv scheduling feature over 802.11 is required to enable the robotic use case in a shared network with best effort IIoT traffic. The results also show that the current 802.1Qbv implementation over 802.11 requires careful planning and configuration of the protected window sizes to account for all channel access, transmission, and software stack overheads. It was also observed that the TCP transport imposes some challenges when used in real-time applications and with TSN features, as the network schedule must consider TCP ACKs that are transmitted by the receiving node, therefore introducing additional overhead in terms of larger protected windows. Future work is needed to design more efficient schedules that accommodate TCP. Moreover, a UDP transport may be considered a more efficient option without loss of reliability as the deterministic delivery is ensured by the TSN layer; however, research is required to demonstrate this assertion. Finally, we have demonstrated that TSN is an effective tool for assuring the operational performance of a latency-sensitive control application while maximizing bandwidth utilization for other IIoT devices.

Disclaimer

Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

References

[1] Martinez, B., Cano, C., and Vilajosana, X. (2019). A Square Peg in a Round Hole: The Complex Path for Wireless in the Manufacturing Industry. IEEE Communications Magazine, 57(4), 109–115. https://doi.org/10.1109/MCOM.2019.1800570

[2] Y. Liu, M. Kashef, K. B. Lee, L. Benmohamed and R. Candell, "Wireless Network Design for Emerging IIoT Applications: Reference Framework and Use Cases," in Proceedings of the IEEE, vol. 107, no. 6, pp. 1166-1192, June 2019.

[3] IEEE 802.1 Time Sensitive Networking (TSN) Task Group: https://1.ieee802.org/tsn/.

[4] M. Eisen, M. M. Rashid, A. Ribeiro, and D. Cavalcanti. "Scheduling Low Latency Traffic for Wireless Control Systems in 5G Networks." arXiv preprint arXiv:1910.13587 (2019).

[5] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008) , vol., no., pp.1-499, 16 June 2020, doi: 10.1109/IEEESTD.2020.9120376.

[6] S. Sudhakaran, K. Montgomery, M. Kashef, D. Cavalcanti and R. Candell, "Wireless Time Sensitive Networking Impact on an Industrial Collaborative Robotic Workcell," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2022.3151786.

[7] "iPerf - The ultimate speed test tool for TCP, UDP and SCTP," https://iperf.fr/iperf-doc.php

[8] M. Kashef, Y. Liu, K. Montgomery, and R. Candell, "Wireless Cyber-Physical System Performance Evaluation Through a Graph Database Approach," in ASME. Journal of Computing and Information Science in Engineering, April 2021, vol. 21, no. 2.

[9] Y. Liu, R. Candell, M. Kashef, and K. Montgomery, "A collaborative workcell testbed for industrial wireless communications—the baseline design," 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), pp 1315–1321, June 2019.

[10] D. Ginthör, J. von Hoyningen-Huene, R. Guillaume and H. Schotten, "Analysis of Multi-user Scheduling in a TSN-enabled 5G System for Industrial Applications," 2019 IEEE International Conference on Industrial Internet (ICII), 2019, pp. 190-199, doi: 10.1109/ICII.2019.00044.

[11] K. Montgomery, R. Candell, Y. Liu, and M. Kashef, "Wireless User Requirements for the Factory Workcell," Advanced Manufacturing Series (NIST AMS) 300-8.

[12] Stanford Artificial Intelligence Laboratory et al. Robotic Operating System. (2018,5,23), https://www.ros.org

[13] Universal Robots Support, Real-Time Data Exhcange (RTDE) Guide, https://www.universal-robots.com/articles/ur/interface-communication/real-time-data-exchange-rtde-guide/.

[14] R. Candell, "Measurement data from Operational Impacts of IEEE 802.1Qbv Scheduling on a Collaborative Robotic Scenario," National Institute of Standards and Technology, https://doi.org/10.18434/mds2-2811, 2022.