Periodic Control Traffic Support in a Wireless Time-Sensitive Network

Pablo Avila-Campos*, Jetmir Haxhibeqiri*, Xianjun Jiao*, Muhammad Aslam*, Gilson Miranda Jr.†,
Ingrid Moerman*, Jeroen Hoebeke*
*IDLab, Ghent University – imec
†IDLab, University of Antwerp – imec
{pabloesteban.avilacampos, jetmir.haxhibeqiri, xianjun.jiao, muhammad.aslam, ingrid.moerman, jeroen.hoebeke}@ugent.be
gilson.miranda@uantwerpen.be

***Abstract*—Time Sensitive Networking (TSN) is being utilized for industrial deterministic applications. Machine control is an example of such applications, which requires high reliability, low and deterministic latency. Currently, such requirements can only be met by wired networks that do not offer the wireless flexibility. For supporting TSN in end-to-end wired-wireless networks, TSN features need to be presented in the wireless network domain as well. In this demo we show the ability of wireless TSN (W-TSN) to support periodic machine control traffic with low latency under other background traffic in the network. We demonstrate time synchronization and scheduling mechanisms in a wireless setting by employing a control loop (a system to balance a ball in a canal) scenario where the time-critical traffic maintains the required latency under scheduled case. This will be demonstrated in a setup composed of imec's W-TSN evaluation kit built on top of the openwifi SDR platform.**

***Index Terms*—wireless time-sensitive networking, time synchronization, scheduling, openwifi, IEEE802.11.**

I. INTRODUCTION

Time Sensitive Networking (TSN) [1] is a set of standards defining mechanisms for deterministic data transmission over Ethernet based networks. In many industrial use case scenarios, portability and mobility is a must (AGV communication, robot communication, mobile machine control communications etc). As such, wired TSN can only support a subset of the aforementioned use cases as it comes with cabling costs, lack of mobility support and flexibility. However, in many use cases, a combination of wireless flexibility as well as wired TSN determinism is needed to fulfill the use case needs. As such, bringing communication determinism to the wireless network domain is of utmost importance to support end-to-end low-latency deterministic communication in collaboration with wired TSN.

In order to support determinism, any network needs to employ at least two features: accurate time synchronization throughout the network [2] and means to schedule [3] and differentiate traffic flows. In [4] we have presented the accurate time synchronization over wireless links by utilizing the PTP and accurate timestamping of openwifi [5]. Next to μs level time synchronization, traffic scheduling is supported in the W-TSN enabled openwifi boards by means of a gated mechanism that gives channel access to specific queues periodically. As such, deterministic communication for periodic control loop applications can be supported by W-TSN. In [6] authors give key parameters of different industrial traffic types and how communication technology can be chosen based on the traffic type. Periodic control loop traffic type does not require high throughput, however, typically it requires low latency communication with periodicity that ranges from some ms up to several 10s of ms [6].

In [7] authors show how time-aware scheduling in wireless links can offer possibility to support robotic arm communication over wireless. Similarly, we will show the support for periodic control traffic in the W-TSN by employing accurate time synchronization, traffic scheduling and differentiation, as well as real-time monitoring for performance validation.

This demo paper is structured as follows. Section II will describe the design and implementation of building blocks for the demo, section III will describe the demo setup while section IV will give an overview of the demo procedure and expected results. Section V will conclude this paper.

II. DESIGN AND IMPLEMENTATION

In this section we will show the design and implementation needed for the demo. We will describe the data generation, data forwarding and monitoring aspects of the solution.

A. Data Generation

In this demo machine control traffic is generated by means of a proportional–integral–derivative (PID) control loop feedback. The PID control aim is to balance a ball in the middle of a canal using an infrared distance sensor and a stepper motor as actuator. A PID control is defined by three parts: the controller, the sensor, and the process. In this demo, the sensor and the controller are placed in two different Raspberry Pi-s (RPIs). The first RPI is in charge of sampling measurements from the sensor every 10ms, which gives information about the ball position. Then, this information is included in a UDP packet and transmitted to the second RPI through the W-TSN as shown in Figure 2. The second RPI receives the ball position, calculates the error, and applies a PID correction to the canal angle using a stepper motor.

Furthermore, a second traffic flow is produced between both RPI-s to account for other background traffic in the network. This dummy traffic is also UDP traffic, generated using $iperf$,

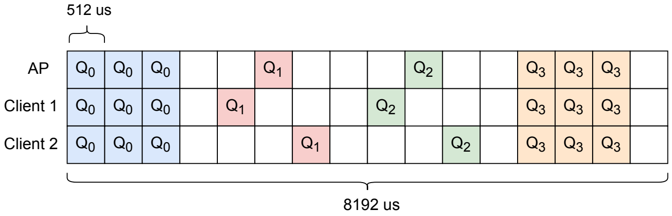

Figure 1. Schedule applied during demonstration

whose function is to show the time multiplexing capabilities of the demo.

B. Data Forwarding

To support deterministic communication we have implemented the accurate time synchronization over wireless [4], achieving down to 1 μs accuracy. In addition to the time synchronization, traffic scheduling is achieved by implementing a gated mechanism in each node that will give timely access to each hardware queue of the node [8]. As such, dedicated channel access can be given to certain queues of a certain node in a given time to account for the required latency and reliability of the traffic flow.

Time sensitive networks are originally layer 2 networks where data is forwarded at layer 2. We kept the same concept for the W-TSN, by offering the data forwarding on layer 2 between different wireless nodes. As such, the W-TSN will behave as layer 2 bridge between the two RPI-s.

Each wireless node has an schedule for each queue. The schedule consist of a cycle length of 8.192 ms, with 16 time slots of 512 μs each. The distribution of the time slots inside the cycle for each queue in each node is shown in Figure 1. All PTP and wireless control traffic is assigned to queue 0. Time critical traffic (PID traffic) is assigned to queue 1, background traffic ($iperf$ traffic) is assigned to queue 2 while queue 3 is used for publishing the monitored information from the end nodes to the central controller (CNC). The PID control traffic is sourced at client 1 with destination client 2. Thus, the time slots of queue 2 on each node (client 1, AP, client 2) follow each other to give the shortest latency for the traffic. By giving the dedicate time slots for time critical traffic we expect to have the best latency performance for the time critical traffic. In the second case, time slots of the time critical traffic and best effort traffic are shared, to show the performance in a normal shared WiFi channel.

C. Telemetry

Network performance monitoring is done using In-band Network Telemetry (INT) for wireless networks [9]. INT employs monitoring on a per flow basis, on per-hop and end-to-end fashion. Different wireless parameters are monitored on each hope, such as: RSSI, SNR, channel used, data retransmission, data rate(DR) and Modulation and Coding Scheme(MCS) index. Most importantly, on each network hop the packet is timestamped enabling the calculation of accurate latency on each hop and end-to-end.

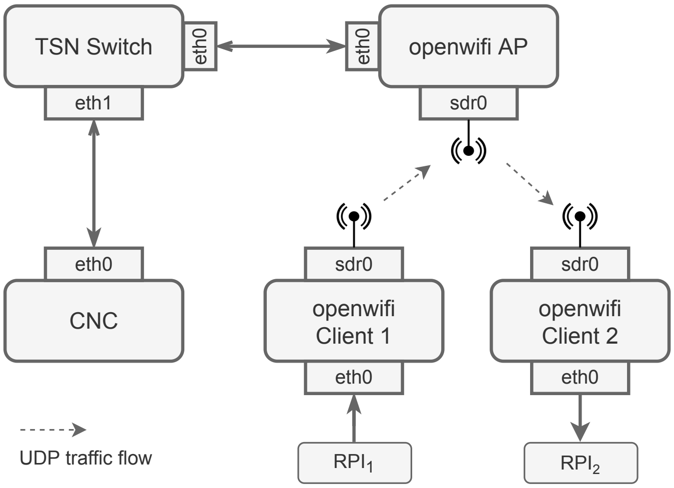

Figure 2. Demo setup architecture

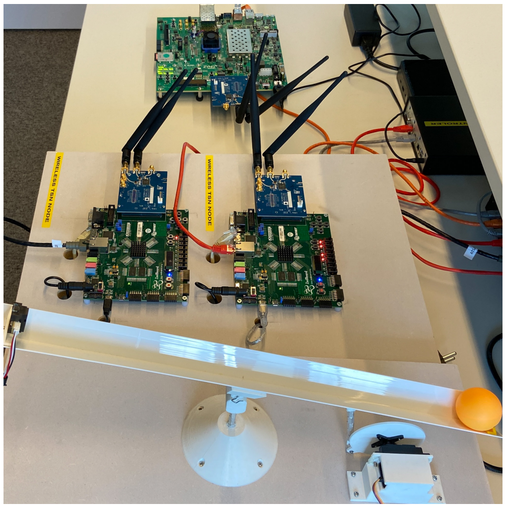

Figure 3. Demo setup picture

III. DEMO SETUP

The described demo is implemented on top of imec W-TSN evaluation kit (EK). imec W-TSN EK uses FPGA-based Software Defined Radio (SDR) platform, $openwifi$ [10] running on Xilinx ZC706 board and Xilinx ZedBoard for the access point (AP) and wireless clients, respectively. Figure 2 presents the demo setup architecture. Each RPI holds part of the PID control loop as described in Section II, while the $openwifi$ clients serve as end nodes of the W-TSN, receiving and transmitting application packets. Additionally, AP forwards packets between the clients and the network. Finally, the Central Network Controller (CNC) is an essential element of a TSN whose function is to provide configuration data, such as the schedule, to TSN nodes in response to application requirements. In this demo, the CNC also receives, monitors and presents the network telemetry data using $Grafana$ dashboard. Finally, the complete demo picture is shown in Figure 3.

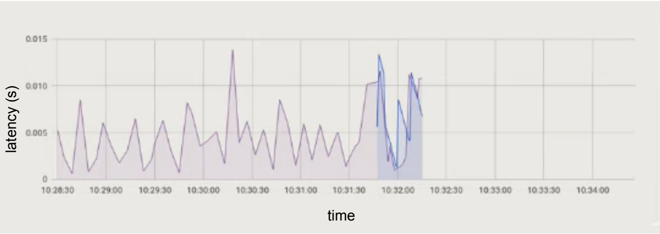

Figure 4. Grafana showing the machine-control and iperf traffic end-to-end latency with dedicated time slot

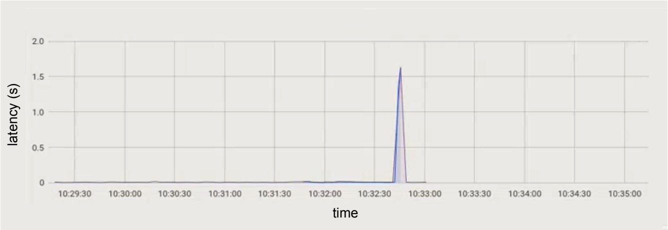

Figure 5. Grafana showing the machine-control and iperf traffic end-to-end latency with shared time slot

IV. MEASURING PROCEDURE AND RESULTS

To demonstrate the capabilities of W-TSN, two experiments are designed. The intention of such experiments is to display the potential of a W-TSN compared to normal Wi-Fi to support time-sensitive traffic, specifically, PID control traffic. Demo visitors can interact by starting/stopping each traffic flow in the network and by managing the network schedule from an user interface from CNC.

A. Dedicated time slot

The first experiment uses the schedule shown in Figure 1. As described in Section II, queue 1 (Q_1) is used by PID control traffic and queue 2 (Q_2) for $iperf$ traffic. With this schedule, both traffic flows have a transmission opportunity every 8192 μs, meeting the time-sensitive application latency requirements. Furthermore, management and telemetry traffic are scheduled in queues 0 and 3 respectively, leaving guard bands to avoid disturbing time-sensitive traffic. Figure 4 depicts the end-to-end latency of machine control traffic (red), and $iperf$ traffic (blue). As seen, both latencies keep under $\sim$ 10 ms. This will be visible also by the stability of the ball in the middle of the canal.

B. Shared time slot

In a non-W-TSN case using CSMA/CA as IEEE802.11, all nodes compete for the channel before transmitting. When high traffic or high node density is present, best-effort networks do not offer latency guarantees. To represent this scenario, both machine control, and $iperf$ traffic are set to share Q_1. The result is shown in Figure 5. As both traffic flows compete for the same time resource, latency is unpredictable any more (rising up to 1.5 s). The effect will also be visible in the balance of the ball. As control traffic is highly time-sensitive, packet delay makes it impossible to keep the ball balanced.

V. CONCLUSION

In this work we demonstrated how a W-TSN overcomes a best-effort network when supporting time critical periodic control traffic. By implementing basic TSN features such as time synchronization and scheduling, the network determinism is improved. Together with determinism, TSN provides low latency and high reliability, key requirements of future industrial networks. On top of $openwifi$, and using a PID control application, this demo shows how our W-TSN can handle multiple traffic flows by using a time-based gating system.

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] N. Finn, “Introduction to Time-Sensitive Networking,” *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 22–28, 7 2018.

[2] “802.1AS-2020 - IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications,” Tech. Rep., 2020.

[3] “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.

[4] 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.

[5] 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.

[6] T.-s. Magyarorsz, “Investigating the network traffic of Industry 4 . 0 applications – methodology and initial results,” *16th International Conference on Network and Service Management (CNSM)*, pp. 1–6, 2020.

[7] S. Sudhakaran, V. Mageshkumar, A. Baxi, and D. Cavalcanti, “Enabling QoS for Collaborative Robotics Applications with Wireless TSN,” *2021 IEEE International Conference on Communications Workshops, ICC Workshops 2021 - Proceedings*, pp. 0–5, 2021.

[8] 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, 11 2021. [Online]. Available: https://link.springer.com/article/10.1007/s11277-021-09056-0

[9] J. Haxhibeqiri, P. H. Isolani, J. M. Marquez-Barja, I. Moerman, and J. Hoebeke, “In-Band Network Monitoring Technique to Support SDN-Based Wireless Networks,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 627–641, 3 2021.

[10] “open-sdr/openwifi: open-source IEEE 802.11 WiFi baseband FPGA (chip) design: driver, software.” [Online]. Available: https://github.com/open-sdr/openwifi