The Quality-aware and Vertical-tailored Management of Wireless Time-Sensitive Networks

Gilson Miranda Jr.*†, Jetmir Haxhibeqiri‡, Nina Slamnik-Kriještorac*, Xianjun Jiao‡,
Jeroen Hoebeke‡, Ingrid Moerman‡, Daniel F. Macedo†, Johann M. Marquez-Barja*
*IDLab - imec, University of Antwerp, Belgium
‡IDLab - imec, Ghent University, Belgium
†Universidade Federal de Minas Gerais - Computer Science Department, Brazil
{gilson.miranda, nina.slamnikkrijestorac, johann.marquez-barja}@uantwerpen.be
{jetmir.haxhibeqiri, xianjun.jiao, jeroen.hoebeke, ingrid.moerman}@ugent.be
damacedo@dcc.ufmg.br

***Abstract*—This paper presents a concept of a TSN Controller NetApp to support diverse vertical applications and provide Quality of Service guarantees during their life-cycle. The NetApp is composed of a Controller entity that receives and processes requests from vertical applications with specific network performance demands, and an Agent entity which applies configurations and monitors the state of network elements. This control architecture has been extended to support wireless TSN communication on top of openwifi, supporting the flexibility required by vertical applications with mobile devices such as drones and automated guided vehicles. We describe the building blocks of the TSNC NetApp supporting wired-wireless TSN deployments and show its experimental results demonstrating the feasibility of our solution.**

***Index Terms*—TSN, Wireless TSN, SDN, Network Management**

I. INTRODUCTION

Emerging technologies such as cloud/edge computing, Artificial Intelligence (AI), digital twins, 5G/6G and WiFi 6/7 networking, are being developed for industrial environments to improve and optimize the performance of industrial processes. Industry 4.0 merges the aforementioned technologies into one ecosystem to enable highly flexible and automated smart factories that can cope with the ever-increasing complexity of manufacturing processes [1]. In such ecosystems, the holistic operation of industrial assets needs to be flexible and automated, as required for future factories. One of the approaches to achieve such holistic operation is to leverage on Internet of Things (IoT), thereby deploying various IoT sensors and actuators to closely monitor and control processes. However, creating a distributed IoT system with multiple devices requires a careful study of communication requirements, which must be satisfied by the network to deliver the expected Quality of Service (QoS) levels for smart factories.

These QoS requirements depend on the type of vertical industry and its applications. In an industrial environment, a diverse range of applications such as process control, video surveillance and object tracking may coexist, each with distinct communication requirements. This coexistence is even more likely in the case of converged Operational Technology (OT) and Information Technology (IT) deployments enabled by Time-Sensitive Networking (TSN) over Ethernet [2]. For instance, process control operations have strict requirements such as bounded low latency of 1-5ms, low jitter (up to 1ms), and reliability in the order of 99,9999 % packet delivery ratio, despite requiring throughput of only a few Mbps in many cases [3]. On the other hand, video-based inspection and surveillance might require throughput of 10-100 Mbps while tolerating up to 1 % of Packet Loss Ratio (PLR) and delays of tens of milliseconds. Moreover, moving devices such as Automated Guided Vehicles (AGVs) and drones require, on top of strict performance requirements, the ability to move around the factory floor, adding more complexity to network planning and management [1].

In this work we present a concept of a TSN Controller NetApp to support diverse vertical applications and provide QoS guarantees during their life-cycle. The NetApp builds on top of a control architecture for TSN networks, leveraging the recent standards for TSN such as time-based traffic shaping, precise time synchronization, and its extensions to the wireless domain provided by the *openwifi* project [4].

II. ENABLERS FOR RELIABLE INDUSTRIAL NETWORKS

To address the stringent requirements of industrial networks, a variety of technologies for industrial communication such as PROFINET, EtherCAT, and Modbus were developed over the last decades, resulting in a fragmented landscape with incompatible protocols [5]. Ethernet TSN is a key enabler for Industry 4.0 due to its support for time-critical, reliable, and deterministic communication, providing a unified communication technology to substitute the legacy and fragmented field bus systems [1]. However, time-critical communication is not restricted to wired networks.

The increasing use of AGVs, drones, and Augmented Reality (AR)/Virtual Reality (VR) applications for worker assistance may greatly benefit from reliable wireless communication. 5G is seen nowadays as the main technology for mobile time-critical applications, however, there is also increasing interest in integrating TSN features into Wi-Fi to support the demands of IoT applications [3]. In comparison to 5G, Wi-Fi networks require simpler infrastructure in terms of software

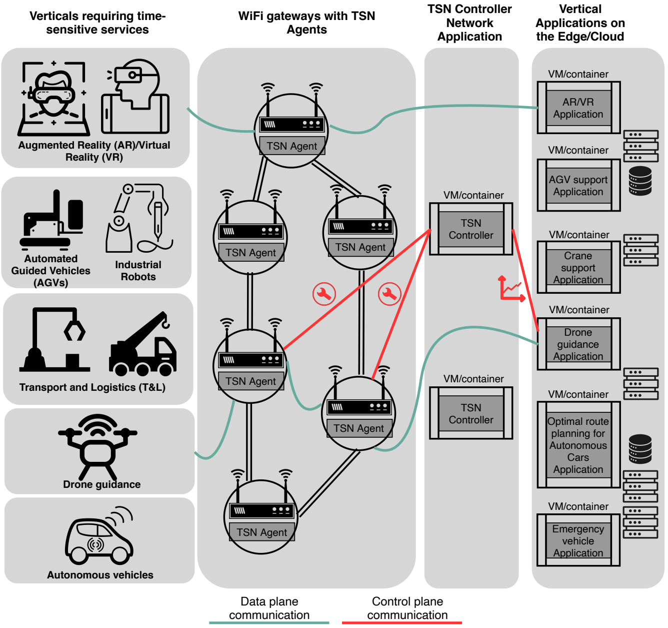


Fig. 1: High-level view on the TSN ecosystem enhanced by TSN Controller NetApp improving QoS for various vertical applications.

and hardware, and can be deployed using unlicensed bands, although with higher probability of competition for spectrum access [5].

In recent years, features such as Orthogonal Frequency-Division Multiple Access (OFDMA), Coordinated Spatial Reuse (CSR), Time-Aware Scheduling (TAS), and Frame Preemption, have been integrated or evaluated for integration into Wi-Fi standards in order to support more reliable communication [3]. The *openwifi* project goes further and implements the functions to achieve precise time synchronization and TAS over Wi-Fi, required for TSN [4]. With the TSN extensions from wired to wireless, it is also crucial to have an effective management framework that binds both domains, enabling proper control and coordination of network resources. Since different verticals have different QoS requirements, in this paper, we propose quality-aware and vertical-tailored mechanisms for efficient management of Wireless TSN (W-TSN) resources.

Network Applications (NetApps) are programmable and flexible pieces of software that interface with the network control plane to improve QoS of vertical applications consumed by end users. Given the emerging popularity of NetApps, in this paper we present a *TSN Controller (TSNC) NetApp* that performs quality-aware management of underlying W-TSN network to boost the performance of applications tailored to vertical industries, i.e., vertical applications. A NetApp can be specified as a fundamental building block of vertical services that simplifies their complex composition and abstracts the underlying network complexity, while bridging the knowledge gap between vertical stakeholders and network experts [6]. In one of the recent works [7], the concept of NetApps has been introduced in the context of various European projects that are currently investigating the potential of accelerating idea-to-market process for verticals such as smart cities and utilities, transportation, automotive, agriculture, energy and e-health, among others. All projects share the same vision of NetApps, i.e., presenting it as a necessary separate middleware layer that is designed to simplify the implementation and deployment of vertical systems. The design of such NetApps follows the cloud-native principles with lightweight software composition and programmable interfaces towards external components, and as such, NetApps could be deployed as virtual machines or containers either on a virtualized network infrastructure, or using vertical-specific hardware.

Given such design principles, in Figures 1 and 2, we present our TSN Controller NetApp. In particular, the TSNC NetApp leverages upon i) the southbound interface for interacting with underlying Network Elements (NEs) (e.g., WiFi gateways), which is used to configure NEs through the TSN agents that apply network configuration changes tailored to expected QoS, and ii) the northbound interface used for collecting vertical-tailored QoS requirements from the vertical applications that are designed and deployed for e.g., AR/VR, AGVs and industrial robots in IoT 4.0 environments, transport and logistics sectors, drone guidance, and autonomous vehicles. In this paper, we further detail on the TSNC NetApp architecture, and its features that are making such a NetApp applicable to various industrial systems, including some of the results that show the potential and feasibility of such NetApp deployments.

III. SOFTWARE-DEFINED NETWORKING FOR TSN

IEEE 802.1Qcc [8] defines interfaces and protocols for administration of TSN networks. It also defines three configuration models for TSN networks with different levels of coordination of the NEs. In all three models, a User/Network Interface (UNI) allows talkers[1] to specify their application characteristics and requirements to the network. Using this information, the bridges are configured to deliver the expected Key Performance Indicators (KPIs) between talkers and listeners [2].

In the **fully distributed** model, the UNI sits between a talker and the bridge it connects to. After the talker makes its request, the bridges propagate the information and perform their configuration locally, without a global network view. This model does not require a Centralized Network Configuration (CNC) entity, however, as bridges have only local information, the configuration might be inefficient. The **distributed user/centralized network** adds a CNC element for centralized control of the network. Requests received by bridges through the UNI are forwarded to the CNC so the path between talker and listener can be properly configured. Lastly, the **fully centralized** model defines the Centralized User Configuration (CUC) entity, which provides a central UNI and is in charge of requesting configuration updates from the CNC on behalf of talkers and listeners.

---

[1] We refer to talkers/listeners and end-nodes interchangeably.

The instantiation and management of TSN networks using Software Defined Networking (SDN) has been investigated in the past years [9], [10]. The proposals focus mostly on wired solutions that provide control functionality for wired field bus protocols and Ethernet-based TSN. Existing solutions still require some operations to be carried offline, demanding manual reconfiguration when network conditions change. This mode of operation increases the time to deploy such networks, while limiting flexibility and increasing costs.

Extending management functions to wireless domains has also been investigated [11]. With the focus on 5G performance, some works tackle the integration of Ethernet TSN with Ultra-Reliable Low-Latency Communication (URLLC), using 5G as a TSN bridge to interconnect two separated TSNs. This way, the TSNC is also involved in the allocation of resources in the wireless 5G domain. However, the scope of such controllers is still limited – the envisioned networks are not completely integrated (e.g., synchronization is still not end-to-end), and they lack features such as automated scheduling and integrated monitoring.

To support the expected flexibility required for industrial scenarios and to build an effective NetApp, the TSNC architecture must integrate the basic building blocks such as monitoring functions, automated control loops, and provide a centralized point of coordination for the NEs. In the following sections, we describe the TSNC proposed in our previous work [12], and extend it to support seamless management of the wireless TSN features of openwifi. Furthermore, we facilitate its deployment and interfacing with vertical applications through a NetApp design.

IV. CONTROLLER NETAPP SUPPORTING WIRED-WIRELESS TSN NETWORKS

In our previous work, we deployed wired TSNs on open testbeds using a prototype architecture and demonstrating its key features for management of such networks [12]. Using *openwifi*, we extended the solution to support W-TSNs and further research the challenges and solutions to effectively manage wired-wireless TSN setups. In this work we further investigate the deployment of the controller architecture as a NetApp, providing the foundation for flexibly supporting vertical industrial applications running on top of TSN-enabled networks. As Figure 1 illustrates, diverse vertical applications are dynamically supported by requesting resources to the TSNC NetApp, which proceeds with the configuration of the NEs and provides network performance guarantees.

The NetApp TSNC architecture in Fig. 2 follows the **fully centralized** configuration model, described in Section III, with the CUC element providing a UNI for the vertical applications. The TSNC NetApp manages NEs such as bridges, end-nodes, and Wi-Fi Access Points (APs) by interacting with a TSN Agent (TSNA) installed on them. Every bridge in the same domain must run the TSNA to allow proper coordination of the network. To facilitate the deployment and configuration of both TSNC NetApp and its TSNAs in dynamic industrial environments, they can be deployed using virtualization methods such as Virtual Machines (VMs) or containers. Bidirectional communication between controller and agents through the southbound interface is necessary, as well as between TSNC NetApp and vertical NetApps through the northbound interface. The overview of these required interfaces is defined in the TSNC NetApp descriptors, which are out of scope of this paper, but they define the relationship/interaction of TSNC NetApp with other components, e.g., TSNAs and/or vertical applications.

While a regular VM is sufficient to deploy the TSNC, the TSNA requires certain hardware support for TSN functionality such as hardware timestamping for accurate synchronization, and multi-queue support for traffic classification and scheduling [12]. Using passthrough techniques it is possible to execute the TSNA using a VM or containers and provide the access to hardware features [13], however, we currently focus on the deployment of TSNAs on bare-metal nodes.

V. MANAGEMENT AND CONTROL FRAMEWORK

The framework is based on a control/agent architecture, which is a common way for abstracting resource management in heterogeneous networks.

A. TSN Controller Architecture

Figure 2 shows the TSNC elements and their main functions. It is composed of five main components and a Management Interface containing an Internal Interface, used for interaction between controller modules, and the UNI. The UNI provides the interfaces for application configuration and performs access control. The CUC is responsible for high-level coordination of network components, collecting and processing the requests from vertical applications and triggering commands to other components of the TSNC. The CNC interacts with the NEs through the TSNA to configure and control NEs. The Scheduler provides routes and traffic shaping rules for each admitted flow. The Monitor collects and stores data-plane monitoring data, which are consumed by the Control Loop to ensure that the expected KPIs are met.

Within the CUC, the Application Registry stores the characteristics and expected KPIs of the running vertical applications. The characteristics comprise packet size, periodicity, inter-packet generation, flow specification (5-tuple of source/destination address, source/destination port, transport protocol), and the KPIs comprise the expected throughput, maximum allowed delay and jitter, and expected reliability in terms of packet delivery ratio. High-level orchestration is performed by the Network Orchestrator (NO). Monitoring rules for In-band Network Telemetry (INT)[2] are derived from the application registry for each new vertical application registered, making the monitoring robust and flexible to handle newly deployed vertical applications as well. Whenever a new configuration for the network is generated, the NO validates the configuration correctness before it is applied to NEs.

---

[2] For more In-band Network Telemetry details we refer the reader to reference [14].

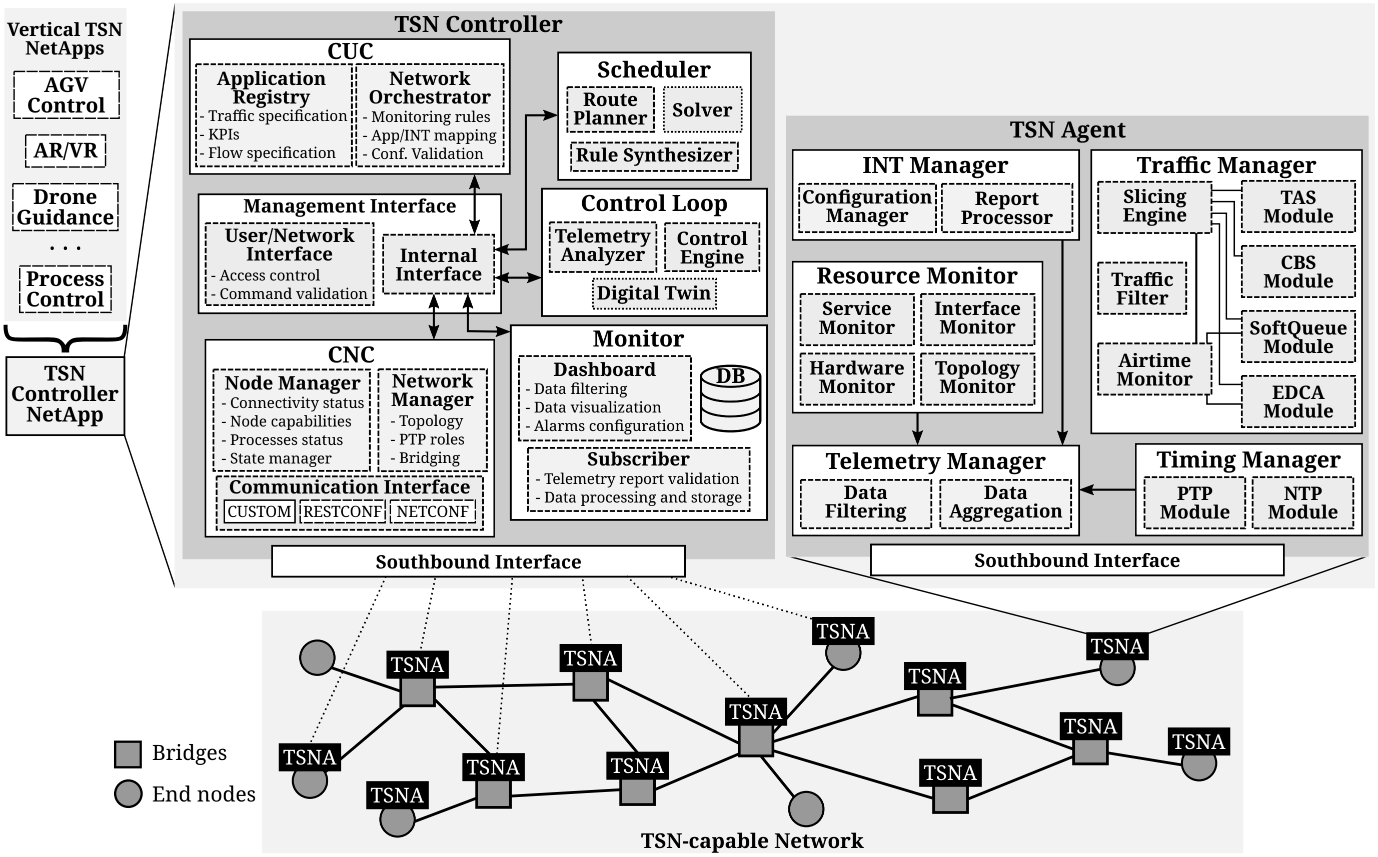


Fig. 2: The Architecture of the TSN Controller NetApp.

Within the CNC, the Node Manager controls and monitors the connectivity of nodes, keeps track of status of relevant processes (e.g., for Precision Time Protocol (PTP) and INT), and holds information about the capabilities of nodes. In the event of a node restart, it restores the node's previous operational state. The Network Manager is responsible for network-level configuration such as bridging/routing, PTP roles (e.g., GrandMaster (GM), boundary-clock, slave), and keeps track of network topology. The communication with NEs is abstracted by the Communication Interface that selects the appropriate communication method for each NE. Currently, the TSNC uses a custom module for communication based on ZeroMQ sockets, however, RESTCONF and NETCONF can also be supported with the appropriate bindings to translate the CNC control commands, which makes the TSNC NetApp more flexible and programmable.

Accurate real-time monitoring is crucial to ensure that the network delivers the KPIs requested by the vertical applications at all times. The Monitor contains a Subscriber to receive telemetry data from NEs, a Database for long-term storage, and a Dashboard for data visualization. The Subscriber module receives the INT reports from NEs, processes them and stores the data into the Database. The Dashboard allows data filtering, statistics visualization, and setting up alarms so the operator is notified when certain events occur.

The Scheduler defines routes and scheduling rules for each bridge in the path between application endpoints. First, a Route Planner defines the appropriate route for the flow and the interfaces which must be affected by the new schedule configuration. Then, a Solver produces configuration to shape the traffic at each interface while keeping the expected KPIs for any other concurrent traffic. The output of the solver is translated by the Rule Synthesizer into a scheduling or traffic-shaping rule for the CNC to apply in the NEs.

After concluding the initial configuration, the Control Loop monitors telemetry data and acts to guarantee that the network keeps providing the required KPIs for each application. The Telemetry Analyzer processes QoS information from each flow and compares them with the requested KPIs. Whenever necessary, the Control Engine acts to adjust the configuration. This component might employ a Digital Twin to estimate and evaluate network changes.

B. TSN Agent Architecture

The TSNA in Fig. 2 manages the configuration of NEs and monitors its resources. It interacts with the CNC, applying the configurations defined by the TSNC services so the network delivers the KPIs required by the vertical applications. It also abstracts configuration details of heterogeneous nodes and notifies relevant events (e.g., topology changes) to the controller. The TSNA is composed of five components. The Timing Manager controls PTP and Network Time Protocol

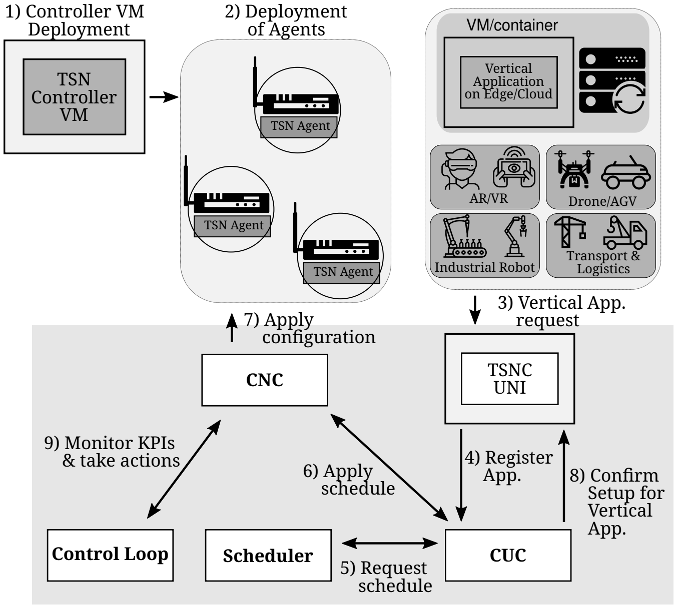


Fig. 3: The operation workflow of TSN Controller NetApp, including its interaction with TSN Agents.

(NTP) and feeds synchronization accuracy statistics to the Telemetry Manager. For network performance verification our solution relies on INT, which allows performance measurements on a per-hop and per-flow basis [5]. The INT Manager component processes the performance reports and centralizes configuration of the INT framework.

The Resource Monitor tracks the status of interfaces, relevant services (PTP, INT framework), topology, and CPU/memory usage. Any changes are reported to the Telemetry Manager, which collects the reports from all components of the TSNA. To reduce signaling traffic in the network, the Telemetry Manager can perform aggregation (e.g., average measurements) and filtering to report only values above a certain threshold.

Lastly, the Traffic Manager controls the traffic shaping features. For this, the Slicing Engine controls modules for TAS, Credit-Based Shaper (CBS), Software-based Queuing (SoftQueue), and Enhanced Distributed Channel Access (EDCA) parameters. The CBS and TAS modules control features already present in Ethernet TSN, with the TAS being also present in openwifi. The SoftQueue and EDCA modules are used in combination with the Airtime Monitor for airtime-based wireless network slicing.

C. TSN Controller NetApp Workflow

For applications with strict communication requirements, all the devices involved must be properly configured in advance [15]. This demands a series of steps, as illustrated in Fig. 3, starting with i) the deployment and configuration of the TSN Controller NetApp itself, and then with ii) the NetApp configuring the network for the vertical time-sensitive services.

The process starts at step (1) with the deployment of a VM with the TSNC, followed by step (2) with the deployment of the TSNAs on bridges and APs. During step (2) the bridges announce their capabilities and resources, and an initial configuration is set by the controller. At this point, the TSN Controller NetApp is ready to receive and process requests to set up end-to-end configurations for vertical time-sensitive applications.

In the step (3), vertical applications make their requests to the UNI informing their QoS requirements and traffic characteristics in terms of maximum data packet size, data transmission periodicity, ports, and protocols. In the step (4) the application is registered in the CUC, which in turn, requests a new network-wide schedule (step (5)) that meets the required QoS for the new application and do not degrade the performance of the already existing time-sensitive flows. The new schedule is validated by the CUC and sent to the CNC on step (6), which applies the configuration to the affected nodes on step (7).

After configuring the network, the CUC is notified and confirms to the vertical application that the network is ready for its operation. An additional step (9) is carried by the control loop, which constantly monitors the network and performs adjustments when necessary to ensure that the admitted vertical applications continue to deliver the requested performance throughout their life-cycle.

VI. EXPERIMENTAL SETUP AND RESULTS

To evaluate the feasibility of the TSN controller NetApp for wired and wireless TSNs, we performed experiments deploying TSN networks on two tesbeds. First, we used the VirtualWall[3], evaluating the resource consumption of the TSNC and its scalability for networks with hundreds of nodes. The experiments were executed with the TSNC running on a *pcgen03-5p* node equipped with Intel Xeon E5645 with 24 threads and 24GB of RAM. Other four similar nodes executed up to 250 TSNAs using docker containers, which connected to the CNC simultaneously. We selected this node for experimentation purposes, but as the results show, the TSNC can also be executed on nodes with lower computing resources. This first set reflects on resource consumption footprint of the TSNC for networks with different amounts of nodes, and enables better TSNC VM dimensioning.

Second, we evaluate the TSNC support for W-TSN and the flexibility of our architecture for a mixed wired-wireless network. To do so, we used the W-TSN Evaluation Kit developed at imec-IDLab, to demonstrate the W-TSN features implemented on top of openwifi and managed by our controller. The evaluation kit, shown in Figure 4, is composed of an Intel NUC operating as CNC, an industrial mini PC equipped with 8 Intel i350 Gigabit Ethernet ports, and a PCEngines APU2 board[4] operating as wired node. The Wi-Fi AP uses a Xilinx ZC102, and the two wireless clients use Xilinx ZedBoard. To perform the experiments, the whole network is synchronized using PTP, and schedules based on

---

[3] https://www.fed4fire.eu/testbeds/virtual-wall/

[4] https://www.pcengines.ch/apu2.htm

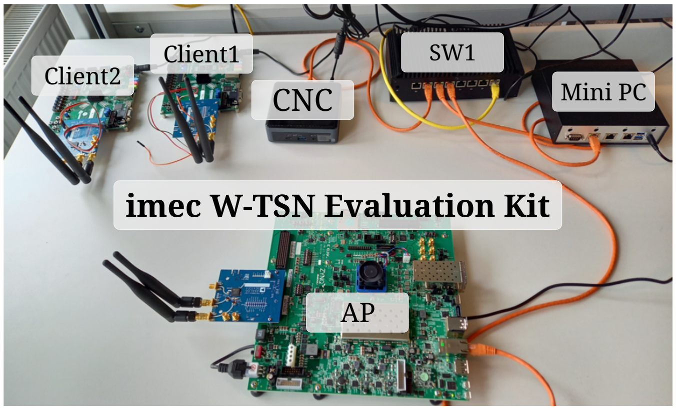


Fig. 4: imec's W-TSN evaluation kit.

the Time-Aware Scheduler of IEEE 802.1Qbv are installed and dynamically modified during the tests.

A. *Results*

Table I shows the CPU, RAM and network loads (Net-In for incoming traffic to the controller, Net-Out for outgoing traffic) of the CNC node of VirtualWall during the process of nodes simultaneously connecting to the controller and receiving their initial configuration. The table shows the results for 100, 500 and 1000 nodes connecting simultaneously. We show the mean and peak usage of resources during the duration of the configuration process by the controller. We observed peaks of 37.5 % of CPU usage and 11.6 Mbps of outgoing network traffic when 1000 nodes connect simultaneously. It should be noted that the connection phase is the most demanding for the CNC in terms of CPU/network load, and that 1000 nodes connecting at the same time is unlikely in most use cases. Nevertheless, after 16 seconds there is no significant CPU activity, and the network load (both incoming and outgoing) is in the order of 50 to 60 Kbps. The maximum RAM usage with all the 1000 nodes connected was 123 MB, which is approximately 123 KB per associated node, including the base memory consumed by the controller itself (approximately 22 MB).

TABLE I: Results of the system load of the TSN Controller NetApp for 100, 500, and 1000 nodes connecting simultaneously.

Nodes	**Metric**	**Resource Usage**			
		CPU	**RAM**	**Net-In**	**Net-Out**
100	**Mean**	4.5 %	32 MB	228 Kbps	1.1 Mbps
	Peak	17.6 %	36 MB	1.5 Mbps	7.5 Mbps
	Duration	4 s	8 s	4 s	4 s
500	**Mean**	12.4 %	59 MB	768 Kbps	4.1 Mbps
	Peak	37.5 %	75 MB	5.3 Mbps	9.0 Mbps
	Duration	7 s	11 s	8 s	8 s
1000	**Mean**	9.2 %	102 MB	710 Kbps	3.5 Mbps
	Peak	37.5 %	123 MB	10 Mbps	11.6 Mbps
	Duration	16 s	25 s	13 s	13 s

These results indicate the low resource footprint of the architecture and its suitability for deployment using VMs with constrained resources. Thus, the deployment of the TSN Controller NetApp can be done faster, as fewer resources must be allocated for the VM, and a single bare metal node can support multiple instances of this NetApp, each controlling distinct domains.

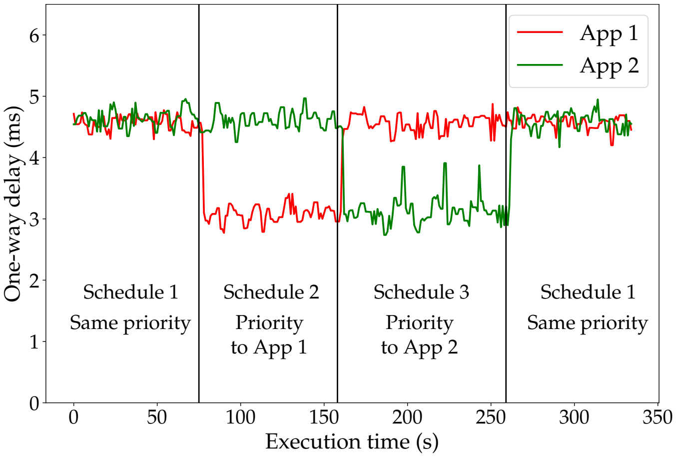


Fig. 5: Impact of traffic scheduling on traffic from the two clients using openwifi TSN extensions.

To demonstrate the capabilities of the TSN Controller NetApp on managing wired and wireless devices of a TSN network and providing quality-aware support for vertical applications, we deployed the NetApp components using the imec-IDLab W-TSN evaluation kit. The nodes synchronize their clocks with the PTP grandmaster, and report the synchronization offset (i.e., error) to the controller. With the openwifi hardware clock extensions for PTP, the 90th percentile of the synchronization offset achieved during the experiments was 2.5μs. Accurate synchronization is crucial for effective scheduling, as it allows us to define slots with lower guard periods – which are silent periods in which the nodes wait without transmitting to make sure that the other nodes are not still configured to the previous slot due to the synchronization offset.

Combined with the extensions for PTP, we leveraged openwifi scheduling capabilities to apply schedules similarly as for wired nodes with IEEE 802.1Qbv TAS. To demonstrate this feature, we managed wireless scheduling in uplink, with each wireless client generating a UDP flow to the wired node. Figure 5 shows the one-way delay of each flow during the experiment execution. We defined three schedules with cycle duration of 5.12 milliseconds, and 20 slots of 256μs of duration. We initially allocated one 256μs to each client, giving their flows the same priority. The graph shows that the one-way-delay for both flows is similar until 75 seconds of execution, when we apply the second schedule.

The second schedule gives an additional slot to client 1, which means one more opportunity to transmit its packets during the cycle, resulting in a lower delay for packets of its flow (App 1). After 160 seconds we apply a third schedule allocating the additional slot to client 2 instead of client 1. We observe the inversion of the flow behavior with App 2 flow (from client 2) achieving lower delay. At 260 seconds the schedules are returned to the initial setting and we observe the same behavior of both flows as in the beginning of the experiment.

By using the recent openwifi TSN extensions, we performed fine-grained control over the traffic performance, even in uplink direction, which requires strict coordination and collaboration of the wireless clients. These results show the capacity of the NetApp to provide efficient management of TSN networks comprising wired and wireless devices, towards providing the QoS guarantees requested by vertical applications.

VII. Conclusion

The diverse network performance demanded by industrial applications requires networks to be flexible and vertical-tailored, thereby applying vertical-specific adjustments towards ensuring required QoS levels. Our TSNC NetApp provides building blocks to coordinate NEs with TSN features, paving the way for flexible and quality-aware management of heterogeneous and flexible TSN network deployments. To realize this concept, our TSNC NetApp gives a central point of coordination for the network, and the interfaces for vertical applications to request resources for their operation. We demonstrated the feasibility of such a Controller NetApp, and evaluated its resource footprint, which indicates its ability to control networks with hundreds of nodes with minimal resource consumption. We also explored the openwifi TSN extensions and showed the ability of the TSNC to perform fine-grained management on traffic behavior for wireless clients. In the current stage, we require the deployment of the TSNAs in bare-metal nodes, in order to access hardware features for TSN operation. In future work we will investigate the deployment using virtualized environments and evaluate the accessible features for even more flexible deployments. Finally, given that such TSNC NetApps can easily interact with vertical applications, as well as network control elements through northbound and southbound interfaces, they can be flexibly deployed on-demand to boost the network performance and improve/maintain service quality taking into account vertical-specific requirements.

Acknowledgment

This research is partially funded by the imec ICON project VELOCe - VErifiable, LOw-latency audio Communication (Agentschap Innoveren en Ondernemen project nr. HBC.2021.0657). This research is also funded by the European Union's Horizon 2020 project VITAL-5G, which is co-funded by the EU under grant agreement No. 101016567. This research is also financed by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and São Paulo Research Foundation (FAPESP) with Brazilian Internet Steering Committee (CGI.br), grants 2018/23097-3 and 2020/05182-3.

References

[1] D. Lou, J. Holler, D. Patel, U. Graf, and M. Gillmore, "The Industrial Internet of Things Networking Framework," pp. 1–70, 2021, online [Available]: https://www.iiconsortium.org/wp-content/uploads/sites/2/2022/08/IINF_Update_2022_08_03.pdf.

[2] L. Lo Bello and W. Steiner, "A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1094–1120, 2019, doi: https://doi.org/10.1109/JPROC.2019.2905334.

[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, 2021, doi: https://doi.org/10.3390/s21154954.

[4] X. Jiao, W. Liu, M. Mehari, M. Aslam, and I. Moerman, "openwifi: a free and open-source ieee802.11 sdr implementation on soc," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–2, doi: https://doi.org/10.1109/VTC2020-Spring48590.2020.9128614.

[5] J. Haxhibeqiri, X. Jiao, E. Municio, J. M. Marquez-Barja, I. Moerman, and J. Hoebeke, "Bringing Time-Sensitive Networking to Wireless Professional Private Networks," *Wireless Personal Communications*, pp. 1–17, 2021, doi: https://doi.org/10.1007/s11277-021-09056-0.

[6] N. Slamnik-Kriještorac, G. Landi, J. Brenes, A. Vulpe, G. Suciu, V. Carlan, K. Trichias, I. Kotinas, E. Municio, A. Ropodi, and J. M. Marquez-Barja, "Network Applications (NetApps) as a 5G booster for Transport & Logistics (T&L) Services: The VITAL-5G approach," in *2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2022, pp. 279–284, doi: https://doi.org/10.1109/EuCNC/6GSummit54941.2022.9815830.

[7] 5GPPP, "Network Applications: Opening up 5G and beyond networks," *5G-PPP Software Network Working Group*, 2022.

[8] "IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks – Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements," *IEEE Std 802.1Qcc-2018*, pp. 1–208, 2018, doi: https://doi.org/10.1109/IEEESTD.2018.8514112.

[9] S. B. H. Said, Q. H. Truong, and M. Boc, "SDN-Based Configuration Solution for IEEE 802.1 Time Sensitive Networking (TSN)," *SIGBED Rev.*, vol. 16, no. 1, p. 27–32, Feb. 2019, doi: https://doi.org/10.1145/3314206.3314210.

[10] T. Kobzan, I. Blöcher, M. Hendel, S. Althoff, A. Gerhard, S. Schriegel, and J. Jasperneite, "Configuration Solution for TSN-based Industrial Networks utilizing SDN and OPC UA," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2020, pp. 1629–1636, doi: https://doi.org/10.1109/ETFA46521.2020.9211897.

[11] S. Bhattacharjee, K. Katsalis, O. Arouk, R. Schmidt, T. Wang, X. An, T. Bauschert, and N. Nikaein, "Network Slicing for TSN-Based Transport Networks," *IEEE Access*, vol. 9, pp. 62 788–62 809, 2021, doi: https://doi.org/10.1109/ACCESS.2021.3074802.

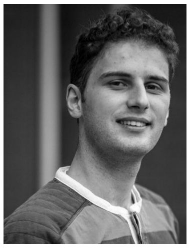
[12] G. Miranda, E. Municio, J. Haxhibeqiri, D. F. Macedo, J. Hoebeke, I. Moerman, and J. M. Marquez-Barja, "Time-Sensitive Networking Experimentation on Open Testbeds," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2022, pp. 1–6, doi: https://doi.org/10.1109/INFOCOMWKSHPS54753.2022.9798073.

[13] V. Maffione, L. Rizzo, and G. Lettieri, "Flexible virtual machine networking using netmap passthrough," in *2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2016, pp. 1–6, doi: https://doi.org/10.1109/LANMAN.2016.7548852.

[14] 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, 2020.

[15] D. Bruckner, M. P. Stanica, R. Blair, S. Schriegel, S. Kehrer, M. Seewald, and T. Sauter, "An Introduction to OPC UA TSN for Industrial Communication Systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1121–1131, 2019, doi: https://doi.org/10.1109/JPROC.2018.2888703.

Gilson Miranda Jr. holds B.Sc. and M.Sc. degrees in Computer Science from Federal University of Lavras (UFLA), Brazil. In 2017 started his PhD in Computer Science at the Federal University of Minas Gerais (UFMG), Brazil. Now is pursuing a Joint PhD in Applied Engineering at the University of Antwerp, Belgium, where he is carrying his research with IDLab. His main research interests are programmable Time-Sensitive Networks, wireless networks, and machine learning applied to network management.



Jetmir Haxhibeqiri received the Masters degree in Engineering (information technology and computer engineering) from RWTH Aachen University, Germany (2013). In 2019, he obtained a Ph.D. in Engineering Computer Science from Ghent University with his research on flexible and scalable wireless communication solutions for industrial warehouses and logistics applications. Currently he is a senior researcher in the Internet Technology and Data Science Lab (IDLab) of Ghent University and imec. His current research interests include wireless communications technologies (IEEE 802.11, IEEE 802.15.4e, LoRa) and their application, IoT, wireless time sensitive networking, in-band network monitoring and wireless network management.

Nina Slamnik-Kriještorac is currently a PhD researcher in the field of Applied Engineering Sciences, at the University of Antwerp and the IMEC research center in Belgium. She obtained her Master degree in telecommunications engineering at Faculty of Electrical Engineering, University of Sarajevo, Bosnia and Herzegovina, in July 2016. In the period from 2016 to 2018, she worked as a Teaching Assistant at University of Sarajevo. She authored or co-authored several publications in journals and international conferences. Her current research is mostly based on NFV/SDN-based network architectures with edge computing for vehicular systems, and the management and orchestration of the flexible and programmable next generation end-to-end network resources and services, with the focus on edge applications.

Xianjun Jiao received his bachelor degree in Electrical Engineering from Nankai university in 2001 and Ph.D. degree in communication and information system from Peking University in 2006. After his studies, he worked in research and product teams in the leading industrial companies of wireless technology, such as Radio System Lab of Nokia Research Center, devices department of Microsoft and Wireless Software Engineering department of Apple. In 2016, he joined IDLab, a core research group of imec with research activities embedded in Ghent University and University of Antwerp. He is working as senior researcher at imec on real-time Software Defined Radio (SDR) platform. His main interests are SDR, signal processing and parallel/heterogeneous computation in wireless communications. On his research track, 30+ international patents/papers have been granted/published.

Jeroen Hoebeke is an associate professor in the Internet Technology and Data Science Lab of Ghent University and imec. He is conducting and coordinating research on deterministic and time-sensitive wireless communication, wireless network management, tighter application-network integration, (industrial) IoT connectivity and embedded communication stacks. This expertise has been applied in a variety of application domains such as logistics, Industry 4.0, building automation, healthcare and animal monitoring. He is particularly active in national funded projects as well as in defining, executing and managing such projects. He has also been involved in several EU research funded projects and is author or co-author of more than 200 publications in international journals or conference proceedings.

Ingrid Moerman received her degree in Electrical Engineering (1987) and the Ph.D. degree (1992) from the Ghent University, where she became a part-time professor in 2000. She is a staff member at IDLab, a core research group of imec with research activities embedded in Ghent University and University of Antwerp. She coordinates the research activities on intelligent Wireless Networking (iWiNe) at Ghent University, where she leads a team of more than 30 researchers. She is also Program Manager of the 'Deterministic Networking' track, part of the CONNECTIVITY program at imec, and in this role she coordinates research activities on end-to-end wired/wireless networking solutions driven by professional and mission-critical applications that have to meet strict Quality of Service requirements. She has a longstanding experience in running and coordinating national and EU research funded projects. She has coordinated several FP7/H2020 projects (CREW, WiSHFUL, eWINE, ORCA). She is involved in many national and H2020 and Horizon Europe projects related to connected vehicles and 5G/6G (Smart Highway, CONCORDA, 5G-MOBIX, 5G-CARMEN, 5G-Blueprint, DEDICAT-6G, HEXA-X II, TrialsNet and 6G-SHINE). She participated in the prestigious DARPA Spectrum Collaboration Challenge (SC2) as lead of Team SCATTER. This team, consisting of researchers from imec-IDLab and Rutgers University (US), has been awarded with two prizes of 750,000 USD each in Phase 1 (2017) and Phase 2 (2018) of the DARPA SC2 competition, and was one of the 10 finalists at the DARPA SC2 championship event organized at Mobile World Congress in Los Angeles (US) in October 2019 (https://www.darpa.mil/news-events/2019-09-10).

Daniel F. Macedo is Professor Associado (equivalent to the Professor level in the US) in the Computer Science Department (DCC) in Federal University of Minas Gerais (UFMG), Brazil. He holds the CNPq research productivity scholarship level 2, a scholarship granted to the most performing researchers in Brazilian academia. He was a post-doc researcher in UFMG, Brazil. He holds a PhD in computer science from Université Pierre et Marie Curie-Paris VI. He also holds a M.Sc and a B.Sc. in Computer Science from UFMG. His main research interests are wireless networks, intelligent management of computer networks,software-defined networking and Internet of Things.

Johann Marquez-Barja is a Professor at the University of Antwerp (Rank 7th in the Times Higher Education Under 50) and also a Professor in IMEC Research Centre (Worldwide leading in Nanotechnologies and Digital solutions), Belgium. Currently, he is leading the Flexible & Programmable Networks Group at IDLab/imec Antwerp. Previously he led the Wireless Cluster. He was/is involved in more than 20 European research projects ranging from long-range to high-capacity radio and networking technologies. He is currently the technical coordinator of the 5G Blueprint project, which focuses on improved 5G cross-border networks to enable the teleoperation of vehicles and vessels. He is a member of IEEE Standards Association, Association for Computing Machinery (ACM), a Fellow of European Association for Innovation (EAI), a Senior Member of the IEEE Communications Society, IEEE Vehicular Technology Society, and IEEE Education Society, where he participates on the board of the Standards Committee. His main research interests are 5G advanced architectures, including edge computing; flexible and programmable 5G and 6G end-to-end networks; IoT communications, and applications. He is also interested in smart mobility and smart city deployments. He leads the Citylab Smart City testbed and the Smart Highway testbed, located in Antwerp, Belgium. He is also active in education development and actively involved in different research actions to enhance engineering education, particularly remote experimentation for online labs. He has given several keynotes and invited talks at various major events, received 30 awards in his career so far, and co-authored more than 200 articles. He is also serving as Editor and Guest editor for different International Journals, as well as participating in several Technical Programme and Organizing Committees for several worldwide conferences/congresses.