Time-Sensitive IIoT Flows over Wi-Fi: a Network Calculus Approach

Carlos Barroso-Fernández, Jorge Martín-Pérez, Constantine Ayimba,
and Antonio de la Oliva *Senior member, IEEE*

***Abstract*—Real time control of connected industry devices such as mobile robots constituting Industrial Internet of Things (IIoT) has made time-sensitive communications over Wi-Fi increasingly important. In order to provide support for time-sensitive services over Wi-Fi, key features of the IEEE 802.11 standard such as restricted Target Wake Time (rTWT) and multi-user Orthogonal Frequency Division Multiple Access (OFDMA) can be exploited. However, even with rTWT and OFDMA, Time-Sensitive Networking (TSN) over Wi-Fi is still a challenge given the unpredictability of wireless channels and the increasing number of Wi-Fi-enabled devices. In this paper, we present a comprehensive network calculus-based analysis of the delay bounds achievable in Wi-Fi networks, leveraging these IEEE 802.11 enhancements alongside synchronized priority queuing via IEEE 802.1Qbv. We propose PONTE, a novel Fully Polynomial Time Approximation Scheme (FPTAS) scheduler that guarantees strict non-violation probabilities (e.g., 99.99%) for inelastic, time-critical traffic over an 802.11 wireless network. Our extensive analysis and simulations in an IIoT scenario demonstrate that PONTE effectively manages dense, mixed traffic flows, achieving TSN objectives with minimal impact on fairness. Our approach is two orders of magnitude faster than the state-of-the-art.**

***Index Terms*—Network calculus, stochastic scaling, optimization, time-sensitive networks, rTWT, Wi-Fi, Industrial Internet of Things**

I. Introduction

The use of collaborative robots (cobots), that operate in the same environment as human workers, has been on the rise recently given the considerable improvements in efficiency they bring to the IIoT. The latter is due to the synergy obtained when routine or repetitive tasks such as pick/place are done by robots (e.g., robotic arms) and unconventional ones are left to humans – see Fig. 1. However cobot deployment is dependent on the operational health and safety standards they can guarantee owing to the risks of injury to humans [1].

This work was supported by José Castillejo mobility grant of the Spanish Ministry of Science, Innovation and Universities (grant no. CAS23/00455). This publication is partially funded by the project 6GINSPIRE PID2022-137329OB-C42, funded by MCIN/ AEI/10.13039/501100011033/; it has been partially funded by the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union-NextGenerationEU through the UNICO 6G I+D 6G-DATADRIVEN, and by the European Commission Horizon Europe SNS JU PREDICT-6G (GA 101095890) Project.

Carlos Barroso-Fernández, Constantine Ayimba and Antonio de la Oliva are with the Department of Telematics Engineering, Universidad Carlos III de Madrid. E-mail: cbarroso@pa.uc3m.es, {ayconsta,aoliva}@it.uc3m.es.

Jorge Martín-Pérez is with Information Processing and Telecommunications Center, and Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid. E-mail: jorge.martin.perez@upm.es.

DOI: 10.1109/JIOT.2025.3623878

Fig. 1. Exemplary IIoT scenario with people and robots collaborating. The Wi-Fi Access Point provides connectivity to the time-sensitive flows of the camera, forklift and robotic arms (highlighted icons).

Typically, the cobot is designed to cease operation when a human is close to it [2].

The safety distance depends on the reaction time of the cobot once a human is detected. The faster the reaction time, the better the safety of workers on the factory floor and the shorter the distance allowed meaning that the robot can keep working uninterrupted for longer in close proximity to humans. Recent advances in AI-aided computer vision and control promise to facilitate this tolerance. Such systems require the use of high capacity compute resources beyond those embedded in the cobot. This creates the need for highly reliable, low latency networks to enable the transmission of video streams and sensor data to the compute resources (ideally on the network edge) and the delivery of control directives back to the cobots.

Moreover, Flexible Manufacturing Systems require reliable, real-time communication that allow for timely updates to connected machines (constituents of the IIoT) to dynamically change production rates in a synchronized manner [3].

In scenarios with fixed/static robots, this is reliably achieved using wired Time-Sensitive Networking (TSN) but in those with mobile cobots performing such tasks as pick and place, wireless TSN over Wi-Fi (IEEE 802.11) networks is the more feasible option [4]–[8]. The latter poses unique challenges given the unpredictability of the Wi-Fi channel and its bandwidth constraints. Video streams (non-TSN flows) and periodic sensor data (TSN flows) from the same cobot need to be properly scheduled in the uplink to ensure that the correct control directives (TSN flows) from the control systems are generated and delivered in the downlink to the cobots in a timely manner.

Recent Wi-Fi amendments – such as 802.11ax, 802.11be and 802.11bn – have introduced key enabler features for TSN. For instance, (i) rTWT to restrict the channel access during given time windows; (ii) 802.11ax OFDMA to have

multiple concurrent transmissions at different frequencies; and (iii) 802.1Qbv to control the time at which queues of different priorities transmit packets. However, the design of a Wi-Fi scheduler for TSN flows (considering packet loss, retransmissions, and prioritization) is still an open challenge.

Gaps in the Literature: Recent studies recognize the complexity of scheduling time-sensitive traffic over Wi-Fi, as detailed in Sec. II. Existing solutions have utilized TWT scheduled channel access, 802.11ax OFDMA scheduling, and prioritization at egress queues to achieve bounded latency. However, the literature often overlooks the necessary jitter tolerance and reliability levels (e.g., 99.99% of packets arrive before the deadline) required for TSN flows. Furthermore, no existing work fully integrates these three key features—rTWT, OFDMA scheduling, and prioritization—into a cohesive solution for TSN over Wi-Fi. To address this gap, we propose PONTE, a Polynomial-based Network Calculus Solution for Time-Sensitive traffic over Wi-Fi. PONTE leverages rTWT scheduled access and 802.11ax OFDMA to *prevent contention* and *ensure reliability*, while also using traffic prioritization and playout buffers to achieve *near-zero jitter*.

PONTE: key features. Our work uses stochastic scaling curves [9] and Network Calculus [10] to determine the reliability (percentage of packets successfully received within a given delay and jitter bound) achieved by a rTWT session in the presence of packet loss, as discussed in Sec. V-C. The obtained reliability bound results in a polynomial whose sign indicates whether the reliability levels are met, taking into account packet errors, retransmissions, and the presence of multiple priority queues within the Stations (STAs). Using this polynomial, PONTE schedules OFDMA transmissions of multiple STAs across the available 802.11ax Resource Units (RUs). Additionally, PONTE achieves near-zero jitter by employing a playout buffer [11, Chapter 9.2] at the Access Point (AP).

Challenges. Though rTWT and OFDMA are key enablers to achieve determinism in Wi-Fi networks, any solution relying on such features has to jointly tackle the following challenges: (i) determine the adequate rTWT session duration and period accounting for traffic requirements of each STA; (ii) schedule packet transmissions of concurrent rTWT sessions accross multiple RUs to avoid contention; (iii) guarantee high reliability, accounting for packet losses and the impact of retransmissions in exacerbating delays and (iv) ensure near-zero jitter regardless of the unpredictability of the wireless channel.

Contributions. Our work advances the state of the art by providing analytical insights into scheduling time-sensitive traffic in the Wi-Fi Uplink (UL). Our key contributions are as follows:

(C1) We devise a reliability bound for traffic scheduled with rTWT using network calculus and stochastic scaling. We propose a procedure that determines whether Wi-Fi time-sensitive traffic meets reliability requirements, given: (i) the number of retransmissions; (ii) traffic profile; and (iii) violation requirement.

(C2) We formulate the joint rTWT and OFDMA scheduling problem for TSN traffic over Wi-Fi, proving its NP-hard nature by reducing it to a classical optimization problem.

(C3) We propose PONTE, an FPTAS algorithm with $(2+\epsilon)$-approximation guarantees and polynomial runtime. The proposed algorithm specifies: (i) different rTWT session durations/periods; (ii) OFDMA scheduling across RUs; and (iii) jitter mitigation through playout buffers.

(C4) We evaluate and validate PONTE in an IIoT scenario with TSN flows or robots, vehicles and interactive video.

To the best of our knowledge, PONTE is the first algorithm using stochastic network calculus – in particular stochastic scaling – to jointly exploit rTWT and OFDMA scheduling for time-sensitive traffic in Wi-Fi, thus guaranteeing strict traffic reliabilities.

The rest of the paper is organized as follows: In Sec. II, we highlight the related work and point out how our solution advances the state-of-the-art. In Sec. III, we briefly introduce the key concepts essential to our approach. In Sec. IV we introduce our model. Thereafter in Sec. V, we derive expressions for arrival and service curves, considering retransmissions due to packet errors. In Sec. VI, we present the joint rTWT and OFDMA scheduling problem and provide proof of its NP-hardness. We provide a detailed description of PONTE in Sec. VII and subsequently in Sec. VIII, present the results of its performance. Sec. X provides conclusions and future research.

II. Related Work

To begin with, we briefly review relevant literature to contextualize and motivate our work. For readers unfamiliar with certain background concepts, we refer them to Section III.

Recent standardization work by the IEEE 802.1 TSN task group resulting in mechanisms such as 802.1AS, 802.1Qbv, 802.1Qch has motivated the research community to devise scheduling solutions to meet the requirements of time-sensitive traffic. Techniques including Cyclic Queuing and Forwarding [12] or Gate Control Lists (GCLs) [13], [14] have been proposed. Heuristics [12], [14] and AI [15] using these standard enhancements to achieve negligible jitter and bounded delay also feature in the literature.

To deliver time-sensitive traffic over Wi-Fi, 802.11 features such as TWT have been proposed to schedule channel access and mitigate contention. Existing works propose deciding the TWT listening interval [16], [17] of STAs, slicing [18], [19] every TWT session among STAs, or using ML for traffic clasification to schedule periodic TWT sessions [20]. Approaches presented in [17], [19] study TWT scheduling to meet the per-packet deadlines, but only [19] is designed for time-sensitive traffic whereas [17] does not consider traffic classes or priorities. However, none of these works assess OFDMA scheduling or provide reliability guarantees.

Works tackling the 802.11ax OFDMA scheduling problem [21]–[24] have as their objective the maximization of the throughput in the Wi-Fi channel. But the proposed solutions do not accommodate time-sensitive traffic and also disregard the traffic reliability. Only recent works as [25], [26] exploit the idea of using 802.11ax OFDMA to schedule time sensitive

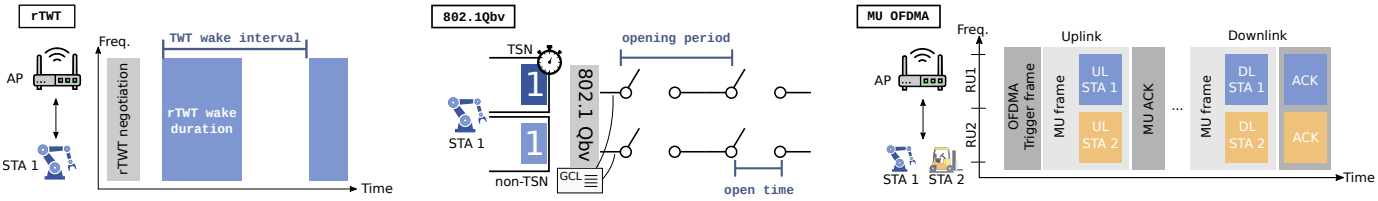

Fig. 2. Schematic of technologies used in this work. With rTWT (left) the AP schedules periodic sessions for the STA to have restricted channel access. 802.1Qbv (middle) periodically opens gates of TSN and non-TSN queues to control packet exit, giving priority to TSN packets. 802.11ax MU OFDMA (right) schedules RUs to different STAs to send traffic at the same time.

traffic. In particular, [25] proposes a scheduling algorithm for haptic control, and [26] just provides guidelines on how to blend OFDMA with TWT to meet time-sensitive requirements. Still, none of the existing works provide a solution that blends OFDMA with rTWT and 802.1Qbv. Moreover, they do not provide scheduling guarantees on delay or reliability.

To the best of our knowledge, the only works that provide delay and reliability guarantees are those scheduling URLLC traffic in 3GPP cellular networks. Concretely, recent works leverage stochastic network calculus [27] to bound the traffic delay of URLLC services [28], [29]. The work in [30] extends this to vehicular networks.

III. Background

In this section we present the IEEE 802.1 and 802.11 features used by PONTE, our proposed scheduler for IIoT TSN flows over Wi-Fi. We then briefly introduce basic concepts of network calculus that are essential to our analysis.

A. Technologies

(r)TWT: The IEEE 802.11be standard [31] introduced a new version of a scheduling system called Target Wake Time (TWT), known as restricted Target Wake Time (rTWT). TWT is a feature that allows Wi-Fi devices (called STAs) to follow a schedule that tells them when to be active and when to sleep – see Fig. 2 (left). This feature was originally designed to save battery life, however, it has also been found useful for improving Wi-Fi performance in crowded networks. Each STA in a TWT session follows two key settings:

- `Minimum TWT wake duration`: the shortest time a STA stays awake before going back to sleep – see the colored boxes from Fig. 2; and
- `TWT Wake interval`: how often the STA wakes up to communicate – the time between the start of the first and second blue boxes from Fig. 2.

These settings are decided through a negotiation between the AP and the STAs.

To further improve TWT, the 802.11be amendment introduces an important feature to protect channel access. Transmissions of rTWT-compliant STAs cannot cross the beginning rTWT wake interval, hence contention is reduced. However: (i) STAs still contend for channel access right after the start of the rTWT wake interval; and (ii) longer rTWT wake intervals do not always result into larger throughput [32].

IEEE 802.1Qbv: IEEE 802.1Qbv, developed by The IEEE 802.1TSN Task Group [33], introduces a scheduling system that prioritizes network traffic by using gates at the egress queues, where outgoing data is processed before transmission – see Fig. 2 (middle). Devices use a GCL, which periodically opens these gates to allow specific types of traffic to be sent at scheduled times. In case gates of different queues are opened concurrently – as in Fig. 2 (middle) – 802.1Qbv first dispatches the high priority packets of e.g. TSN traffic. Priorization and scheduled gate openings ensure that high-priority data is transmitted without delays caused by other network activity. It was revinitially introduced to be applied over 802.3 wired networks, but it can be also applied over any IEEE 802 networks, such as Wi-Fi.

Multi user OFDMA: As noted by [34], OFDMA is a key technology that improves time-sensitive communication in Wi-Fi networks. Introduced in Wi-Fi 6 (802.11ax) [35], MU OFDMA allows multiple STAs to send or receive data at the same time, making network performance more efficient. OFDMA divides the channel into smaller sections called RUs – see how yellow and blue STAs send their packets at the same time in Fig. 2 (right). The number and size of these RUs can be adjusted by the AP depending on the available bandwidth and the number of devices connected.

To manage uplink transmissions, where STAs send data to the AP, the AP first schedules transmissions by sending a Trigger Frame. This frame tells each STA which RU to use, and all scheduled STAs then transmit their data at the same time within a special MU OFDMA frame – see Fig. 2 (right). For downlink transmissions, where the AP sends data to STAs, the process is even trimmer: the AP assigns RUs and sends the data within a single OFDMA frame – see Fig. 2 (right). Even though OFDMA improves network efficiency, it still operates within the conventional Wi-Fi channel access mechanism known as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). This mechanism ensures that STAs listen before transmitting to avoid data collisions. However, the AP can adjust certain parameters to increase its chances of gaining access to the channel when necessary, further optimizing performance [36].

Technologies integration. In this work we integrate rTWT, 802.1Qbv and MU OFDMA to schedule uplink TSN traffic. We exploit the technology integration proposed in [26, Fig. 1] to schedule TSN uplink transmissions in three steps:

1) the AP schedules periodic rTWT sessions for every STA;
2) each STA uses 802.1Qbv at the wireless interface and opens TSN and non-TSN queue gates during the whole rTWT session; and
3) the AP issues OFDMA uplink trigger frames in every

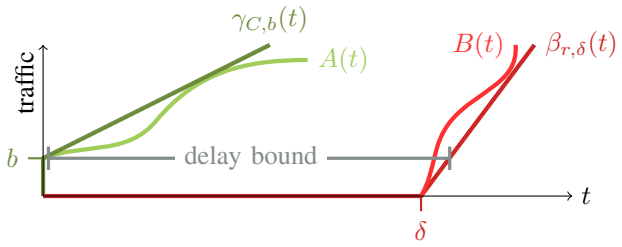

Fig. 3. The arrival and service curves $A(t), B(t)$ are bounded by $\gamma_{C,b}(t)$ and $\beta_{r,\delta}(t)$, respectively. Their horizontal deviation $h(\gamma_{C,b}, \beta_{r,\delta})$ bounds the maximum delay.

rTWT session. The trigger frames assign RUs to STAs transmitting in the rTWT session.

B. Fundamentals of network calculus

Based on [10], we define an arrival curve $A(t)$ to specify the cumulative amount of traffic that arrives at a queue by time t and a service curve $B(t)$ to specify the cumulative amount of traffic served by the Wi-Fi channel by time t. Both curves are wide-sense increasing functions: $A(t') \leq A(t'') \Leftrightarrow t' \leq t''$, analogously with $B(t)$.

In this work we assume known IIoT traffic profiles [13], [37]–[40]. Consequently, we upper bound the arrival curve $A(t)$ with a strict affine arrival curve [10, 1.2.1] $\gamma_{C,b}(t) = Ct+b$. Similarly, we lower bound the service curve $B(t)$ with a strict rate-latency curve [10, 1.3.2] $\beta_{r,\delta}(t) = r[t-\delta]^+$ (where $[x]^+$ is equivalent to $\max\{0, x\}$). Thus, we have $\gamma_{C,b}(t) \geq A(t)$ for all t and $\beta_{r,\delta}(t) \leq B(t)$ for all t.

The experienced delay $d(t)$, resulting from the arrival and service curves, is itself bounded by the horizontal deviation $h(\gamma_{C,b}, \beta_{r,\delta})$

$$d(t) \leq h(\gamma_{C,b}, \beta_{r,\delta}) = \sup_{s \geq 0} \left\{ \inf_{u \geq 0} \{\gamma_{C,b}(s) \leq \beta_{r,d}(s+u)\} \right\} \quad (1)$$

as shown in Fig. 3. Using (1), we obtain the maximum delay of time-sensitive traffic. Notice that delay bound in (1) may go to ∞. We refer the reader to Appendix A for a detailed explanation over this particular case.

IV. System Model

In this section we describe the considered system that we model, an IIoT facility with Wi-Fi TSN flows – see Fig. 4. Then, we mention the working principles of the AP to schedule uplink traffic. Later, we motivate the election of the chosen technologies and, lastly, we introduce the notation used in our system throughout the rest of the paper.

System description. Our system models the uplink traffic exchanged in an IIoT scenario formed of an AP and STAs – see Fig. 4. Each IIoT device corresponds to an STA exchanging TSN and non-TSN wireless uplink traffic with the AP. The STAs have dedicated 802.1Qbv queues at the wireless interface to prioritize TSN traffic over non-TSN traffic. All STAs in the system are rTWT, 802.1Qbv and 802.11ax compliant.

Working principles. We now explain how the AP works in the considered system leveraging Fig. 4 illustration. First, the

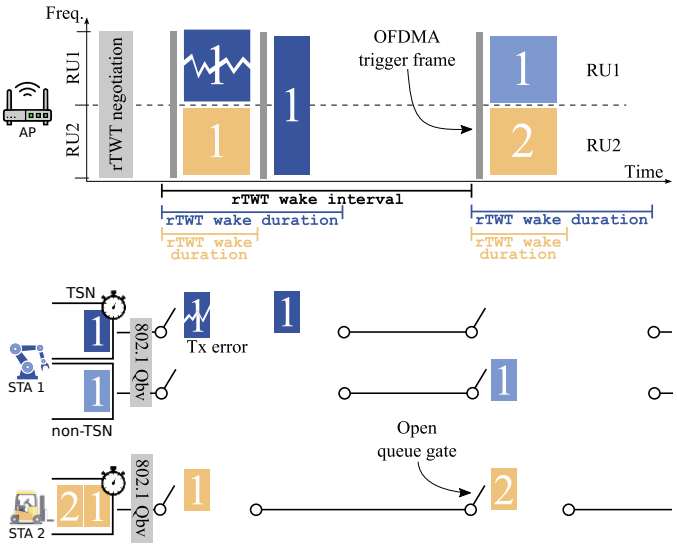

Fig. 4. Example of the considered system: the robot arm and forklift use Wi-Fi to send uplink data to the AP in scheduled rTWT sessions. Both devices use IEEE 802.1Qbv to (i) synchronize gate openings at queues with rTWT; and (ii) prioritize TSN traffic against non-TSN. The robotic arm first TSN transmission is corrupted due to a poor channel in the RU1 band. It is retransmitted successfully within the wake duration.

AP negotiates and schedules periodic rTWT sessions for the robotic arm and forklift. Second, the robotic arm and forklift set the period and opening time of *all* their 802.1Qbv gates to match the rTWT session parameters. In Fig. 4 we observe how the `TWT wake duration` of each STA matches the 802.1Qbv gate opening time, and the `TWT wake interval` matches the 802.1Qbv gate opening periods. Third, the AP sends OFDMA trigger frames during rTWT sessions to schedule MU transmissions so the robotic arm and forklift transmit in separate RUs.

The robotic arm dispatches packets of its 802.1Qbv queues upon MU transmissions. Even though its gates (of TSN and non-TSN traffic) are both opened, 802.1Qbv prioritizes the transmission of packets from the TSN queue – see the dark blue packet 1 in Fig. 4. If there is a transmission error in the TSN packet – as in Fig. 4 –, the packet is retransmitted in the next scheduled MU transmission. Hereof, the transmission of the non-TSN packet (light blue) is delayed by the retransmission of the TSN packet and is transmitted at the last rTWT session illustrated in Fig. 4.

Technologies motivation. The motivation behind using rTWT in the considered scenario is two-fold: (i) it prevents that other STAs contend for channel access during a rTWT session with TSN flows; and (ii) it serves as a mechanism to align 802.1Qbv gate openings with periodic rTWT sessions. The motivation behind using OFDMA is to allow concurrent packet transmissions across RUs and mitigate contention. The motivation behind using 802.1Qbv is to ensure TSN flows have priority over non-TSN flows – thus meeting TSN latency requirements.

Note that solutions using only OFDMA use Enhanced Distributed Channel Access (EDCA) among different Access Categories (ACs) – e.g. TSN and non-TSN traffic. Thus, non-TSN traffic may win the contention and transmit even if there

TABLE I
NOTATION

Symbol	Description
Q_u	List of queues of STA u.
L_u	Minimum wake duration of STA u.
T_u	Doze time of STA u.
δ_u	Time until the first wake period of STA u.
$x_{u,r}$	Whether STA/user u is assigned to RU r.
$p_{u,r}$	Profit of assigning STA u to RU r.
C	Rate of the channel.
N	Number of maximum retransmissions.
W	Maximum timeout for a packet.
p	Packet loss probability.
C_q	Arrival rate at queue q.
$C_q^{(Tot)}$	Total arrival rate at queue q with RTX.
b_q	Maximum burst at queue q.
$b_q^{(Tot)}$	Maximum total burst at queue q with RTX.
$A_q(t)$	Traffic arrival curve at queue q.
$\gamma_{C,b}(t)$	Strict affine arrival curve with rate C and burst b.
$\beta_{r,\delta}(t)$	Strict rate-latency service curve: rate r, latency δ.
$D_q(t)$	Traffic departing curve at queue q.
$l^q_{\max}$	Maximum packet size at queue q.
$l_{\max}$	Maximum packet size.
$\alpha_q^{(i)}(t)$	rTWT strict affine arrival curve of queue q, $\leq i$ RTXs.
$\alpha_q^{(Tot)}(t)$	rTWT strict affine arrival curve of queue q.
$\beta_q^{(i)}(t)$	rTWT strict service curve of queue q for $\leq i$ RTXs.
$\beta_q^*(t)$	rTWT strict service curve of queue q.
$S(\pi)$	Packet transmission scaling curve.
$S^{\hat{\varepsilon}}(\pi)$	Packet transmission stochastic scaling curve.
$\hat{\varepsilon}$	Violation probability of the stochastic scaling curve.
$d_q(t)$	Experienced delay at queue q.
j_q	Experienced jitter at queue q.
$h(\alpha_q^{(Tot)}, \beta_q^*)$	Delay bound.
Δ_q	Delay requirement of queue q.
J_q	Jitter requirement of queue q.
$1-\varepsilon_q$	Reliability requirement of queue q.

is queued TSN traffic. That is, with only OFDMA we cannot guarantee prioritizing TSN flows over non-TSN traffic when a STA transmits in a RU. Hence, we cannot ensure TSN latency requirements. Lastly, we also remark existing solutions [17] based on plain broadcast TWT suffer from packet contention. Hence, plain TWT solutions cannot guarantee latency nor reliability requirements.

Notation. We now describe the notation used throughout the paper, which is summarized in TABLE I. With u we refer to an IIoT device/STA with a list of queues Q_u. For queues are governed by 802.1Qbv, we use $q' \prec q$ to denote a queue $q' \in Q_u$ with TSN traffic has priority over another queue with non-TSN traffic $q \in Q_u$.

We resort to network calculus to model the arriving and service traffic at each queue $q \in Q_u$ – recall Sec. III-B. We use $A_q(t)$ to denote the arriving traffic at queue q by time t. To bound the arriving traffic we use strict affine arrival curves $\gamma_{C,b}(t)$ of rate C and maximum burst of size b. Concretely, the rTWT strict affine arrival curve of each queue is $\alpha_q^{(Tot)}(t) = \gamma_{C_q^{(Tot)}, b_q^{(Tot)}}(t)$, being $C_q^{(Tot)}, b_q^{(Tot)}$ the total arrival rate and burst size entering queue q considering packet retransmissions. We use $\alpha_q^{(i)}(t)$ to refer to the strict affine arrival curve at a queue counting packets with $\leq i$ retransmissions.

With $B_q(t)$ we denote the served traffic at queue q by time t. To bound the served traffic we use strict rate-latency service curves $\beta_{r,\delta}(t)$ with r, δ the rate and latency of the service curve. Specifically, the rTWT strict service curve $\beta_q^*(t)$ at queue q is a strict rate-latency service curve with rate and latency parameters specified later on in Corollary 1. Again, we use $\beta_q^{(i)}(t)$ to refer to the rTWT strict service curve offered to packets at their $\leq i$ retransmission.

The traffic departing at each queue is referred as $D_q(t)$, and we denote the maximum packet size at each queue as $l_q^{\max}$. Hence, we say $l^{\max}$ is the maximum packet size among all queues.

Every queue has different traffic requirements by means of delay Δ_q, jitter J_q and reliability $1-\varepsilon_q$. In particular, the delay $d_q(t)$ and jitter j_q experienced by each queue must be smaller than Δ_q and J_q, respectively. In this work we leverage the network calculus horizontal deviation $h(\alpha_q^{(Tot)}, \beta_q^*)$ to upper bound the experienced delay at each queue q. Corollary 3 provides an expression for the delay bound.

The delay experienced by TSN flows depends on the scheduled periodic rTWT sessions, hereof the 802.1Qbv gate openings. We use L_u to refer to the `Min TWT wake time` and denote with T_u the doze time of an STA. Consequently, $L_u + T_u$ is precisely the `TWT wake interval`. Additionally, we use δ_u to know the first wake instant of an STA with a scheduled periodic rTWT session.

As shown in Fig. 2 (right), to perform OFDMA transmissions the AP schedules RUs to different STAs with overlapping rTWT sessions. In the OFDMA trigger frame the AP specifies which RU to use, hereafter we denote with $x_{u,r} = 1$ that STA u is scheduled RU r during the uplink transmission. Similarly, we denote with $p_{u,r} > 0$ the profit achieved if STA u transmits at RU r. Given the OFDMA frequency assignment, we denote with C the data rate offered to an STA.

Lastly, we remark that packet transmissions may not succeed despite the scheduled channel access in the considered system. In particular, packet transmission may fail due to channel impairment, thereby leading to $\leq N$ retransmissions. We use p to denote the packet loss probability and W to denote the time it takes to realize a packet is lost (e.g. ACK timeout), hence to put a packet back in the queue for retransmission. To capture the probabilistic nature of transmission errors, in this work we use the scaling process $S(\cdot)$ to denote that $S(\pi)$ out of π packets are successfully transmitted – see Sec. V for further details. Moreover, we use the stochastic scaling curve $S^{\hat{\varepsilon}}(\pi)$ to upper bound the scaling process $S(\pi)$. That is, to claim that $S^{\hat{\varepsilon}}(\pi)$ out of π packets are successfully transmitted with probability $1 - \hat{\varepsilon}$.

V. RTWT DELAY MODEL WITH NETWORK CALCULUS

This section models the rTWT delay and reliability through network calculus and stochastic scaling. We first obtain the rTWT service curves $\beta_q^*(t)$ in Sec. V-A. Then, Sec. V-B

presents the rTWT arrival curves $\alpha_q^{(Tot)}(t)$ accounting for Wi-Fi retransmissions. Finally, Sec. V-C combines the service and arrival curves to determine the reliability bound of traffic scheduled with rTWT.

A. rTWT Service Curves

The departure rate of any periodic slotted process (e.g., rTWT, Time Division Multiple Access (TDMA) or 802.1Qbv) can be represented using the following service curve [41]

$$B_{L,T}(t) = C \cdot \max\left\{ \left\lfloor \frac{t}{T+L} \right\rfloor L,\ t - \left\lceil \frac{t}{T+L} \right\rceil T \right\} \quad (2)$$

with C being the transmission rate, L the length of the transmission opportunity and $T+L$ the duration of the period. In (2) the left/right term of the max operator represents the served traffic when t falls within/outside the transmission oportunity.

To simplify the operations, we use the lower bound of (2) which can be expressed as

$$\beta_{L,T,\delta}(t) = \frac{C \cdot L}{T+L}[t-\delta]^+ \quad (3)$$

with δ the first time instant at which traffic is served in a periodic slotted process with service curve $B_{L,T}(t)$.

We use (3) to lower bound the service curve offered by a rTWT session with `wake duration` L, transmission rate C, and a `wake interval` of length $T+L$ that happens for the first time at time δ. That is, $\delta =$ `17.033 ms` means the first wake interval of the rTWT session occurs at `17.033 ms`.

From (3) we know the service rate offered by a rTWT session is lower bounded by $C\frac{L}{T+L}$. If just one 802.1Qbv queue opens its gates during the whole rTWT `wake duration`, its service rate is also lower bounded by $C\frac{L}{T+L}$. The latter holds for it is the gate alignment assumption taken in both Sec. III-A and Sec. IV. However, if *multiple* 802.1Qbv queues with different priorities open their gates during the whole rTWT `wake interval`, the service rate that rTWT offers to a 802.1Qbv is not necessarily lower bounded by $C\frac{L}{T+L}$. In Corollary 1 we derive the service curve that a rTWT session offers to every 802.1Qbv queue that opens its gate during the whole `wake interval`. The result is an extension of [42] and shows the service rate offered by rTWT to a 802.1Qbv queue is decremented by the arriving rate of higher priority traffic from other 802.1Qbv queues.

Corollary 1 (rTWT service curve)**.** *Consider a STA u that opens the gates of all 802.1Qbv queues during the whole rTWT* `wake duration`*. The rTWT session has parameters L_u, T_u, δ_u. If each queue $q \in Q_u$ has a strict affine arrival curve γ_{C_q,b_q}, then it has a strict service curve*

$$\beta_q^*(t) = \left[\left(R_u - \sum_{q' \prec q} C_{q'} \right) t - \left(R_u \delta_u + l_{\max} + \sum_{q' \prec q} b_{q'} \right) \right]^+ \quad (4)$$

with $R_u = \frac{CL_u}{T_u+L_u}$, C the Wi-Fi transmission rate and $l_{\max} = \max_{q'' \succ q}\{l_{max}^{q''}\}$ the max packet size of lower priority queues. Note that (4) *is a strict rate-latency service curve. The left parenthesis is the rate offered to a STA u, which is impacted by the rate of queues of higher priority. The right parenthesis is the delay of the rate-latency curve, that increases based on the maximum packet size and burst of higher priority queues.*

Proof. We mimic the proof of [10, Proposition 1.3.4]. Take the affine lower bound of queue q as the rate-latency function (3). Consider s' the start of the busy period, that is, $s' < s$ with s the start of the backlog period. In the interval $(s', t]$, queue q departing traffic $D_q(t)$ – see Fig. 5 – satisfies:

$$D_q(t) - D_q(s') \geq \beta_q(t-s') - l_{\max} - \sum_{q' \prec q} D_{q'}(t) - D_{q'}(s') \quad (5)$$

Notice that for any queue q, by definition, $D_q(t) - D_q(s') \geq 0$ due to they are wide-sense increasing functions, and $D_q(s') = A_q(s')$, given that at s' there is no backlog. Then, for the strict arrival curve for higher priority queues q':

$$\begin{aligned} D_{q'}(t) - D_{q'}(s') &= D_{q'}(t) - A_{q'}(s') \\ &\leq A_{q'}(t) - A_{q'}(s') \leq \gamma_{C_{q'},b_{q'}}(t-s') \end{aligned} \quad (6)$$

Now, we plug the above inequality in (5) to obtain

$$\begin{aligned} D_q(t) - D_q(s') &= D_q(t) - A_q(s') \\ &\geq \left[\beta_q(t-s') - l_{\max} - \sum_{q' \prec q} \gamma_{C_{q'},b_{q'}}(t-s') \right]^+ \end{aligned} \quad (7)$$

Unraveling $\beta_q(t)$ and $\gamma_{C_{q'},b_{q'}}(t)$ we obtain

$$\begin{aligned} D_q(t) &\geq A_q(s') \\ &+ \left[R_u \left[(t-s') - \delta_u \right]^+ - l_{\max} - \sum_{q' \prec q} (C_{q'}(t-s') + b_{q'}) \right]^+ \end{aligned} \quad (8)$$

Depending on the difference $(t-s')$, (8) changes. If $(t-s') \leq \delta_u$, we have

$$D_q(t) \geq A_q(s'), \quad \forall (t-s') \leq \delta_u \quad (9)$$

With $(t-s') > \delta_u$, we have $[(t-s') - \delta_u]^+ = (t-s') - \delta_u$ and, ordering the expression in function of $(t-s')$, (8) becomes

$$\begin{aligned} D_q(t) &\geq A_q(s') + \left[\left(R_u - \sum_{q' \prec q} C_{q'} \right)(t-s') \right. \\ &\quad \left. - \left(R_u \delta_u + l_{\max} + \sum_{q' \prec q} b_{q'} \right) \right]^+ \\ &= A_q(s') + \beta_q^*(t-s'), \quad \forall (t-s') > \delta_u \end{aligned} \quad (10)$$

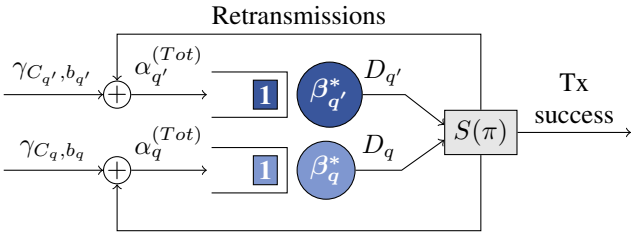

Fig. 5. Diagram flow of Sec. IV delay model. Top and bottom queues q', q have high and low priority, respectively. Each queue has a service curve β_q^* that accounts for aligned 802.1Qbv and rTWT schedules. The departing traffic D_q traverses a scaling process $S(\pi)$ that captures transmission errors (retransmissions) and success. Retransmissions are added to the incoming traffic at each queue γ_{C_q,b_q}, and yields an aggregated arrival curve $\alpha_q^{(Tot)}$.

We now prove whether $D_q(t) \geq A_q \otimes \beta_q^*(t)$ holds. First, we compute the min-plus convolution [10] of $A_q(t)$ and $\beta_q^*(t)$:

$$\begin{aligned} A_q \otimes \beta_q^*(t) &= \inf_{0\leq s'\leq t}\{A_q(s') + \beta_q^*(t-s')\} \\ &= \inf_{0\leq s'<t-\delta_u}\{A_q(s') + \beta_q^*(t-s')\} \\ &\quad \wedge \inf_{t-\delta_u\leq s'\leq t}\{A_q(s') + \beta_q^*(t-s')\} \\ &= \inf_{0\leq s'<t-\delta_u}\{A_q(s') + \beta_q^*(t-s')\} \wedge \inf_{t-\delta_u\leq s'\leq t}\{A_q(s')\} \\ &= \inf_{s':(t-s')>\delta_u}\{A_q(s') + \beta_q^*(t-s')\} \wedge \inf_{s':(t-s')\leq\delta_u}\{A_q(s')\} \end{aligned} \tag{11}$$

with $\wedge$ denoting the minimum operator.

Second, we claim the following about $D_q(t)$, the departing traffic

$$\begin{aligned} D_q(t) &\geq \sup_{s':(t-s')>\delta_u}\{A_q(s') + \beta_q^*(t-s')\} \vee \sup_{s':(t-s')\leq\delta_u}\{A_q(s')\} \\ &\geq \inf_{s':(t-s')>\delta_u}\{A_q(s') + \beta_q^*(t-s')\} \wedge \inf_{s':(t-s')\leq\delta_u}\{A_q(s')\} \\ &= A \otimes \beta_q^*(t) \end{aligned} \tag{12}$$

with $\vee$ the maximum operator and the first inequality holding given (9) and (10).

Overall, we can tell that the departing flow at the queue q satisfies $D_q(t) \geq A_q \otimes \beta_q^*(t)$, with $\beta_q^*(t)$ being the rate-latency service curve. Hence, network calculus concludes that $\beta_q^*(t)$ is a strict service curve for $D_q(t)$. □

Remark: In Sec. III-A, Sec. IV and Corollary 1 we assume *all* gates of 802.1Qbv queues open during the whole rTWT `wake duration`. If we remove such assumption, the service rate offered by rTWT to a unique queue within STA u would be lower bounded by $w_q C \frac{L_u}{T_u+L_u}$. Being $w_q \in [0,1]$ the fraction of time the gate of the 802.1Qbv queue q is opened within the rTWT `wake duration`. Thus, Corollary 1 proof and the following Corollaries should be revisited accordingly to consider the impact of higher priority queues. In particular, the R_u term in Corollary 1 proof should be queue dependent and replaced by $R_{q,u} = w_q R_u$.

B. rTWT Arrival Curves with Wi-Fi Retransmissions

We now present Corollary 2 to derive the affine arrival curve of an STA using rTWT. To this end, We consider that $S(\pi)$ out of π packet transmissions fail and lead to retransmissions – see Fig. 5. Following the approach in [9], [43], we define the scaling process $S(\pi) = \sum_{i=1}^{\pi} X_i$, $X_i \overset{\text{iid}}{\sim} \text{Bernoulli}(p)$, where p represents the packet loss probability. The Bernoulli distribution allows us to reduce the scaling process to an affine curve. In fact, using [9, Theorem 1] we state that the number of failed transmissions, $S(\pi)$, is bounded by the stochastic scaling curve

$$S^{\hat{\varepsilon}}(\pi) = p\pi + 1 - \hat{\varepsilon} \tag{13}$$

with probability $1-\hat{\varepsilon}$ and the following holds:

$$\mathbb{P}\left(\sup_{0\leq a\leq b}\left\{S(a) - S(b) - S^{\hat{\varepsilon}}(b-a)\right\} \leq 0\right) \geq 1-\hat{\varepsilon} \tag{14}$$

In other words, the number of failed transmissions $S(\pi)$ is smaller than $S^{\hat{\varepsilon}}(\pi)$ with probability $1-\hat{\varepsilon}$. We remark from now on we use $S^{\hat{\varepsilon}}(\pi)$ instead of $S(\pi)$ to derive the proof of Corollary 2 and subsequent expressions. The rationale is two-fold: (i) its simplicity; and (ii) its probabilistic guarantee on the packet loss bound.

Corollary 2 (rTWT aggregated arrival curve)**.** *Consider a STA u that opens the gates of all 802.1Qbv queues during the whole rTWT* `wake duration`*. The rTWT session has parameters L_u, T_u, δ_u; and queue $q \in Q_u$, with a strict affine arrival curve γ_{C_q,b_q}. Take a Wi-Fi channel with up to N retransmissions, and packet loss bounded by $S^{\hat{\varepsilon}}(\pi) = p\pi + 1 - \hat{\varepsilon}$. The aggregated arrival curve at queue q is the affine curve*

$$\alpha_q^{(Tot)}(t) = \gamma_{C_q^{(Tot)},b_q^{(Tot)}}(t) = \sum_{j=0}^{N} \gamma_{p^j C_q, b_{q,j,\infty}}(t) \tag{15}$$

with $C_q^{(Tot)} = \sum_{j=0}^{N} p^j C_q$ and $b_q^{(Tot)} = \sum_{j=0}^{N} b_{q,j,\infty}$ the total aggregated rate/burstiness, and

$$b_{q,j,\infty} = p^j C_q \sum_{i=1}^{j} T_{i,\infty} + p^j b_q + (1-\hat{\varepsilon}) \sum_{i=0}^{j-1} p^i + j p^j C_q W \tag{16}$$

being the burstiness associated to the j^{th} retransmission, where W is the Wi-Fi timeout to issue a retransmission. Additionally, $T_{i,\infty} = \frac{\det(A_i)}{\det(A)}$ [1], where A_i is the matrix A with the i^{th} column replaced by the vector $\phi = (\phi_1, \ldots, \phi_N)$. Matrix A and vector ϕ are defined as:

$$A = \begin{pmatrix} d_1 & a_2 & a_3 & \ldots & a_{N-1} & a_N \\ a_2 & d_2 & a_3 & \ldots & a_{N-1} & a_N \\ a_3 & a_3 & d_3 & \ldots & a_{N-1} & a_N \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_N & a_N & a_N & \ldots & a_N & d_N \end{pmatrix} \tag{17}$$

[1]Here $\det(A)$ denotes the determinant of the matrix A.

$$d_j = R_u - \sum_{q' \prec q} C_{q'}^{(Tot)} - 2C_q \sum_{i=j}^{N} p^i \tag{18}$$

$$a_j = -C_q \sum_{i=j}^{N} p^i \tag{19}$$

$$\phi_j = R_u \delta_u + l_{\max} + \sum_{q' \prec q} b_{q'}^{(Tot)} + b_q \sum_{i=j}^{N} p^i + (1-\hat{\varepsilon}) \sum_{k=j-1}^{N-1} \sum_{i=0}^{k} p^i + C_q W \sum_{i=j}^{N} i p^i \tag{20}$$

with $R_u = \frac{CL_u}{T_u + L_u}$ the transmission rate of the rTWT session, C the Wi-Fi transmission rate and $l_{\max} = \max_{q'' \succ q}\{l_{\max}^{q''}\}$ the max packet size of lower priority queues. Overall, the rTWT aggregated arrival curve (15) *is an affine curve whose rate and burst size corresponds to the sum of original packet transmissions and retransmissions.*

Proof. We base our proof on the results presented in [9], which models the arrival curve at a queue using a wireless channel with packet losses. As in [9], we denote with $\alpha_q^{(j)}(t)$ the arrival curve for packets coming from queue q that are retransmitted j times – hence $\alpha_q^{(0)} = \gamma_{C_q,b_q}$. With $\beta_q^{(i)}(t)$ we refer to the service curve that rTWT offers to i^{th} retransmissions at queue q. Note we use i, j to refer to retransmission indices.

Having this in mind we first proceed and derive the service curve for packets that have been retransmitted i times. Then, we derive the arrival curve $\alpha_q^{(j)}(t)$ of packets that have been retransmitted j times. Afterwards, we explain how to compute $\alpha_q^{(j)}(t)$. Lastly, we explain how to compute the total aggregated arrival curve $\alpha_q^{(Tot)}(t)$.

Service curve for i^{th} retransmissions $\beta_q^{(i)}(t)$. As in [9], we assume a packet that is retransmitted for the third time has priority over: a packet that is transmitted for the first time, or a packet that is retransmitted for the second and first time. Consequently, the service curve offered for packets retransmitted i times is $\beta_q^{(i)}(t) = [\beta_q^*(t) - \sum_{j=i+1}^{N} \alpha_q^{(j)}(t)]^+$. In other words, the service curve offered for i^{th} retransmissions is decremented by the arrival rate of $i+1, i, \ldots, N$ retransmissions.

Arrival curve for j^{th} retransmissions $\alpha_q^{(j)}(t)$**.** We now compute in three steps the arrival curve of packets that require the j^{th} retransmission at queue q – termed as $\alpha_q^{(j)}(t)$. First, we recall from [10] that $(\alpha_q^{(0)} \oslash \beta_q^{(0)})(t)$ specifies[2] the amount of original traffic ($j = 0$ retransmissions) that traverse the wireless channel.

Second, we remark that some packets may fail their wireless transmissions due to channel impairment. To consider lost packets we leverage the stochastic scaling curve (13), which bounds the amount of lost packets. Thus, we bound the amount of original transmissions that need to be retransmitted for the first time with $S^{\hat{\varepsilon}}((\alpha_q^{(0)} \oslash \beta_q^{(0)}))(t)$.

Third, we account for the fact that the STA having queue q will take W time units to realize a packet transmission failed. To capture that, we use the burst-delay function [10]:

$$\delta_W(t) = \begin{cases} 0, & t \le W \\ \infty, & \text{otherwise} \end{cases} \tag{21}$$

Concretely, using $S^{\hat{\varepsilon}}((\alpha_q^{(0)} \oslash \beta_q^{(0)})) \oslash \delta_W(t)$ we obtain the arrival curve of packets coming back to queue q for their $j = 1$ retransmission. Note the deconvolution with $\delta_W(t)$ delays by W time units the entrance of the first retransmissions to the queue q.

We now generalize how to compute the arrival curve of packets requiring a j^{th} retransmission. Concretely, we obtain

$$\beta_q^{(0)} = \left[\beta_q^*(t) - \sum_{j=1}^{N} \alpha_q^{(j)}\right]^+, \quad \alpha_q^{(0)} = \gamma_{C_q,b_q}$$

$$\beta_q^{(1)} = \left[\beta_q^*(t) - \sum_{j=2}^{N} \alpha_q^{(j)}\right]^+, \quad \alpha_q^{(1)} = S^{\hat{\varepsilon}}(\alpha_q^{(0)} \oslash \beta_q^{(0)}) \oslash \delta_W$$

$$\beta_q^{(2)} = \left[\beta_q^*(t) - \sum_{j=3}^{N} \alpha_q^{(j)}\right]^+, \quad \alpha_q^{(2)} = S^{\hat{\varepsilon}}(\alpha_q^{(1)} \oslash \beta_q^{(1)}) \oslash \delta_W$$

$$\ldots$$

$$\beta_q^{(N)} = \beta_q^*(t), \qquad \alpha_q^{(N)} = S^{\hat{\varepsilon}}(\alpha_q^{(N-1)} \oslash \beta_q^{(N-1)}) \oslash \delta_W$$

and substituting the expressions for $\beta_q^{(i)}(t)$ we obtain the general expressions for $\alpha_q^{(j)}(t)$:

$$\alpha_q^{(0)} = \gamma_{C_q,b_q}$$

$$\alpha_q^{(1)} = S^{\hat{\varepsilon}}\left(\alpha_q^{(0)} \oslash [\beta_q^* \oslash \alpha_q^{(1)} \oslash \alpha_q^{(2)} \oslash \cdots \oslash \alpha_q^{(N)}]^+\right) \oslash \delta_W$$

$$\alpha_q^{(2)} = S^{\hat{\varepsilon}}\left(\alpha_q^{(1)} \oslash [\beta_q^* \oslash \alpha_q^{(2)} \oslash \alpha_q^{(3)} \oslash \cdots \oslash \alpha_q^{(N)}]^+\right) \oslash \delta_W$$

$$\ldots$$

$$\alpha_q^{(N)} = S^{\hat{\varepsilon}}\left(\alpha_q^{(N-1)} \oslash [\beta_q^* \oslash \alpha_q^{(N)}]^+\right) \oslash \delta_W.$$

In the above expressions we should use the rTWT service curve $\beta_q^*(t)$ specified in Corollary 1. Concretely, we should use Corollary 1 considering traffic arriving at a queue is the addition of original flows and packet retransmissions – as shown in Fig. 5. That is, in Corollary 1 statement we must replace $\gamma_{C_{q'},b_{q'}}(t)$ by $\alpha_{q'}^{(Tot)}(t)$ for high priority queues – hence $C_{q'}, b_{q'}$ by $C_{q'}^{(Tot)}, b_{q'}^{(Tot)}$. The expressions for the latter terms are given at the last step of our proof.

Computing $\alpha_q^{(j)}(t)$. We now explain how to compute the arrival curve of packets requiring the j^{th} retransmission. In the expressions above we see $\alpha_q^{(j)}(t)$ is self-dependant for it appears in both the left and right-hand side of the equality. Thus, we have to solve a fixed point iteration problem that – according to [9] – reduces to solving a system of equations presented in (22).

Note at queue q the rate of packets that come for a j^{th} retransmission is $p^j C_q$. Thus, to compute the arrival rate of j^{th} retransmissions $\alpha_q^{(j)}(t) = \gamma_{p^j C_q, b_{q,j,\infty}}(t)$ all we need to know is its burst size – termed as $b_{q,j,\infty}$ inline with [9] notation. According to [9, Sec. IV.C.2)], the burst size is (16).

[2]Min-plus deconvolution is defined as $f \oslash g(t) = \sup_{u \ge 0}\{f(t+u) - g(u)\}$.

To compute the $T_{i,\infty}$ terms present in the burst size expression in (16), we have to solve the following system of equations

$$A \times (T_{1,\infty}, T_{2,\infty}, \ldots, T_{N,\infty})^t = \phi \tag{22}$$

with the matrix A and vector ϕ defined in [9, Sec. IV.C.2)] replacing R by $R_u - \sum_{q' \prec q} C_{q'}^{(Tot)}$ and RT by $R_u\delta_u + l_{\max} + \sum_{q' \prec q} b_{q'}^{(Tot)}$. Note, $C_q^{(Tot)}, b_q^{(Tot)}$ are the rate and burst of the total aggregated rate, given in the last step of our proof.

The above modifications on A, ϕ are because [9] does not consider an rTWT service curve but a rate-latency curve $\beta(t) = R[t-T]^+$. Nevertheless, the rTWT service curve $\beta_q^*(t)$ is also a rate-latency service curve with rate $R_u - \sum_{q' \prec q} C_{q'}^{(Tot)}$ and latency term $(R_u\delta_u + l_{\max} + \sum_{q' \prec q} b_{q'}^{(Tot)})/(R_u - \sum_{q' \prec q} C_{q'}^{(Tot)})$.

We solve the system in (22) following the approach in [9, Sec. IV.C.2)] Specifically, we use the Cramer's Rule and get $T_{i,\infty} = \det(A_1)/\det(A)$, with A_i being the matrix that replaces with ϕ the column i of matrix A. As a result, the expression for $\alpha_q^{(j)}(t)$ is precisely the one presented at the end of [9, Sec. IV.C.2)]. That is, we have $\alpha_q^{(j)} = \gamma_{p^j C_q, b_{q,j,\infty}}(t)$ with $b_{q,j,\infty}$ the expression presented in (16).

Computing $\alpha_q^{(Tot)}(t)$. To conclude the proof, we recall $\alpha_q^{(Tot)}(t)$ is the addition of original traffic and packet retransmissions at queue q. Concretely, we have

$$\alpha_q^{(Tot)}(t) = \sum_{j=0}^{N} \alpha_q^{(j)}(t) = \sum_{j=0}^{N} \gamma_{p^j C_q, b_{q,j,\infty}}(t) \tag{23}$$

Since the addition of affine arrival curves is [10] an affine arrival curve whose rate and burst size is the sum of the additive arrival curves, we have $\sum_{j=0}^{N} \gamma_{p^j C_q, b_{q,j,\infty}}(t) = \gamma_{\sum_j p^j C_q, \sum_j b_{q,j,\infty}}$. The latter is precisely the aggregated arrival curve presented in (15), with $C_q^{(Tot)} = \sum_{j=0}^{N} p^j C_q$ and $b_q^{(Tot)} = \sum_{j=0}^{N} b_{q,j,\infty}$.

□

Remark: To compute the aggregated arrival curve $\alpha_q^{(Tot)}(t)$ of queue q, we first need to compute the arrival curves of higher priority queues $q' \prec q$. Note $C_{q'}^{(Tot)}, b_{q'}^{(Tot)}$ appear in (18) and (20), respectively.

Remark II: It is worth mentioning that in order to apply Corollary 2, it is mandatory to meet the stability condition claimed in [9, (4)], which in our case translates to:

$$R_u - \sum_{q' \prec q} C_{q'}^{(Tot)} > C_q^{(Tot)} \tag{24}$$

In other words, the service rate offered by the rTWT service curve $\beta_q^*(t)$ (left side), should be greater than the total arrival rate (right side).

C. rTWT Delay/Reliability Bounds with Wi-Fi Retransmissions

Given the rTWT service and arrival curves in Corollaries 1 and 2, we now resort to network calculus to bound the *delay* and *reliability* experienced by STAs using rTWT. In particular, we compute the delay and reliability bounds considering packet retransmissions, which are bounded by the stochastic scaling curve $S^{\hat{\varepsilon}}(\pi)$ presented in (13).

To compute the maximum *delay* experienced by an STA using rTWT we leverage the horizontal deviation [10] between the service $\beta_q^*(t)$ and arrival curve $\alpha_q^{Tot}(t)$ – as depicted in Fig. 3. We now present Corollary 3, which provides a bound of the maximum delay experienced by an STA given its rTWT parameters.

Corollary 3 (rTWT delay bound)**.** *Consider a STA u that opens the gates of all 802.1Qbv queues during the whole rTWT* `wake duration`*. The rTWT session has parameters L_u, T_u, δ_u; and queue $q \in Q_u$, with a strict affine arrival curve γ_{C_q,b_q}. Take a Wi-Fi channel with up to N retransmissions, and packet loss bounded by $S^{\hat{\varepsilon}}(\pi)$. The delay experienced by queue q is upper bounded by $h(\alpha_q^{(Tot)}, \beta_q^*)$ with probability larger than $(1-\hat{\varepsilon})^N$, being*

$$h(\alpha_q^{(Tot)}, \beta_q^*) = \frac{b_q^{(Tot)} + R_u\delta_u + l_{\max} + \sum_{q' \prec q} b_{q'}^{(Tot)}}{R_u - \sum_{q' \prec q} C_{q'}^{(Tot)}} \tag{25}$$

Intuitively, the delay bound (25) *is the time it takes to transmit the maximum burst size (numerator) given the rate offered by the rTWT session (denominator).*

Proof. First, we note that the strict service curve $\beta_q^*(t)$ for queue q is a rate-latency service curve [10] given by Corollary 1. In other words, we express the strict service curve as $\beta_q^*(t) = r[t-\delta]^+$ with

$$r = R_u - \sum_{q' \prec q} C_{q'}^{(Tot)}, \quad \delta = \frac{R_u\delta_u + l_{\max} + \sum_{q' \prec q} b_{q'}^{(Tot)}}{R_u - \sum_{q' \prec q} C_{q'}^{(Tot)}} \tag{26}$$

Second, we remark that traffic arriving at queue q is upper bounded by the affine arrival curve [10] $\alpha_q^{(Tot)}(t) = C_q^{(Tot)}t + b_q^{(Tot)}$ specified in Corollary 2. Both its terms unroll as

$$C_q^{(Tot)} = \sum_{j=0}^{N} p^j C_q^{(0)}, \qquad b_q^{(Tot)} = \sum_{j=0}^{N} b_{q,j,\infty} \tag{27}$$

Lastly, we recall [10, 1.4.1], the delay bound considering an affine arrival curve and a rate-latency strict service curve is $\frac{b_q}{r} + \delta$. Hence, substituting (27) in (26), leads to (25).

To conclude, we proceed as described in [9, (5)] to ensure (25) is satisfied with probability $(1-\hat{\varepsilon})^N$. That is, (25) holds if the bound for the number of packet errors $S^{\hat{\varepsilon}}(\pi)$ is satisfied in each of the N retransmissions. □

Note that the preceding analysis only considers successfully transmitted/re-transmitted packets. To obtain the reliability of rTWT we must compute the percentage of packets that arrived on time, considering that lost packets do not arrive on time. In Corollary 4 we specify the confidence that a packet is successfully transmitted within the delay bound from Corollary 3.

Corollary 4 (rTWT reliability bound)**.** *Consider an STA scheduled with rTWT in the conditions stated in Corol-*

lary 3. Any packet of queue q is successfully transmitted in $h(\alpha_q^{(Tot)}, \beta_q^*)$ seconds with probability larger than

$$\left[1 - p^{N+1}\right](1 - \hat{\varepsilon})^N \tag{28}$$

The above probability is precisely the chances of having a successful transmission in $N+1$ attempts (left term), and the probability of satisfying the delay requirement (right term).

Proof. We consider the joint probability of a successful transmission and having a delay $d(t)$ smaller than $h(\alpha_q^{(Tot)}, \beta_q^*)$:

$$\begin{aligned}\mathbb{P}\left(\text{Tx success} \wedge d(t) \leq h(\alpha_q^{(Tot)}, \beta_q^*)\right) \\ = \left[1 - \mathbb{P}(N \text{ Tx errors})\right] \mathbb{P}\left(d(t) \leq h(\alpha_q^{(Tot)}, \beta_q^*)\right) \\ \geq \left[1 - p^{N+1}\right](1 - \hat{\varepsilon})^N \end{aligned} \tag{29}$$

The last equality holds given the stochastic scaling curve $S^{\hat{\varepsilon}}(\pi)$ and Corollary 3. □

VI. Wi-Fi Time Sensitive Scheduling Problem

In this section we formulate the problem of scheduling time-sensitive traffic with rTWT and 802.11ax OFDMA. For each STA, we *determine* the adequate rTWT parameters: (i) `TWT Wake interval`, L_u; (ii) `Minimum TWT wake duration`, $L_u + T_u$ and (iii) first session offset δ_u. Additionally, for each OFDMA transmission we must *decide* which STA u is assigned to each RU r, which we denote as $x_{r,u} \in \{0, 1\}$.

The decisions $L_u, T_u, \delta_u, x_{u,r}$ must be made such that the delay Δ_q, jitter J_q, and reliability $1 - \varepsilon_q$ constraints are met for the traffic present in each STA queue, i.e. $\forall q \in Q_u$. Note ε_q refers to the violation tolerance for both the delay and jitter constraints of traffic in queue q.

Problem 1 (rTWT & OFDMA Scheduling)**.** *Consider STAs u with queues Q_u. Each queue $q \in Q_u$ has delay Δ_q, jitter J_q and reliability $1-\varepsilon_q$ requirements. Decide the RU assignment $x_{u,r}$ and rTWT setup L_u, T_u, δ_u to meet traffic requirements, and maximize the scheduling profit.*

$$\max_{x_{u,r}, L_u, T_u, \delta_u} \sum_{r \in R} \sum_{u} x_{u,r} \cdot p_{u,r} \tag{30}$$

$$s.t.: \sum_{u} x_{u,r} \cdot \frac{L_u}{T_u + L_u} \leq 1, \quad \forall r \in R \tag{31}$$

$$\mathbb{P}\left(d_q(t) \leq \Delta_q\right) \geq 1 - \varepsilon_q, \quad \forall t, u, q \in Q_u \tag{32}$$

$$\mathbb{P}\left(j_q \leq J_q\right) \geq 1 - \varepsilon_q, \quad \forall u, q \in Q_u \tag{33}$$

where j_q is the real jitter experienced by queue q and $p_{u,r}$ the profit of assigning STA u to RU r.

Constraint (31) ensures that the fraction of time of rTWT sessions does not exceed 100%, while (32) and (33) ensure that delay and jitter constraints are met with the given reliability level $1 - \varepsilon_q$.

Note the profit values $p_{u,r}$ are defined by the owner of the Wi-Fi AP device. For example, if STAs u, u' have time-senstive and best effort traffic, respectively, setting $p_{u,r} > p_{u',r}, \forall r$ prioritizes time-sensitive traffic over best effort traffic. In Sec. VIII, we use the following profit function

$$p_{u,r} = 1 + \max\left\{\theta \max_{q \in Q_u}\{C_q^{(Tot)} l_{max}^q\}, -\theta / \min_{q \in Q_u}\{\Delta_q\}\right\} \tag{34}$$

to trade-off high throughput traffic ($\theta > 0$) against time-sensitive traffic ($\theta < 0$). Concretely, if $\theta > 0$ the profit is proportional to the product of the rate and maximum packet size among all queues. Whereas if $\theta < 0$ the profit is inversely proportional to the tightest delay requirement among all queues. We remark that Problem 1 is complex and cannot be solved easily – as detailed in the next proposition.

Proposition 1. *Problem 1 is NP-hard.*

Proof. To prove that Problem 1 is NP-hard, we show that it can be an instance of the Generalized Assignment Problem (GAP) [44, (10.7)], which is NP-hard [45].

Consider a problem instance in which the violation tolerance is $\varepsilon_q = 1, \forall q$. Hence, delay (32) and jitter (33) constraints are always met. Assume also that rTWT parameters L_u, T_u, δ_u are given for a sufficiently large number of STAs $u_1, \ldots, u_M$. Also consider that the AP can only split the bandwidth into two RUs r, r' in OFDMA transmissions. Lastly, assume STAs rate requirements are much smaller than that provided by each RU, i.e. $C_r \gg C_q^{(Tot)}, \forall q$.

Clearly, the solution of the instance detailed above consists in just taking OFDMA decisions $x_{u,r}, \forall u$ to set STAs at either of the RUs, i.e. $x_{r,u} + x_{r',u} \leq 1, \forall u$. Thus, the described instance of Problem 1 is an instance of the GAP with: STAs u as items with weights $\frac{L_u}{L_u + T_u}$; RUs r, r' as bins with capacity 1; and profits $p_{u,r}$.

□

VII. PONTE

In this section, we propose PONTE, a solution to Problem 1 that exploits 802.1Qbv, multi user OFDMA transmissions and rTWT. PONTE stands for **PO**lynomial-based **N**etwork calculus solution for **T**ime senstiv**E** traffic. In the following we detail step by step how PONTE solves Problem 1, a schematic of which is depicted in Fig. 6.

Step 1: *rTWT/802.1Qbv period specification.* PONTE sets the `TWT Wake interval` $L_u + T_u$ for every STA u, and fixes the periodicity of 802.1Qbv GCL to match $L_u + T_u$ – i.e. PONTE aligns rTWT and 802.1Qbv. Specifically, PONTE specifies a common period $L_u + T_u = \mathcal{T} = \min_q\{\Delta_q/2\}, \forall u$ that is half the strictest delay requirement among all time-sensitive flows.

Step 2: *ensure rTWT reliability*. PONTE finds the adequate stochastic scaling curves $S^{\hat{\varepsilon}_q}(\pi)$ to ensure the rTWT reliability – see Corollary 4 – meets the requirement $1-\varepsilon_q$ of every queue q. Hence, PONTE sets $\hat{\varepsilon}_q$ of the stochastic scaling curve to ensure the rTWT reliability (28) is greater than/equal to the time-sensitive reliability requirement, i.e. it sets

$$\hat{\varepsilon}_q = 1 + \log_N \frac{1 - \varepsilon_q}{1 - p^{N+1}} \tag{35}$$

Step 3: *rTWT/802.1Qbv session duration.* Third, PONTE specifies the rTWT `Minimum TWT wake duration` L_u

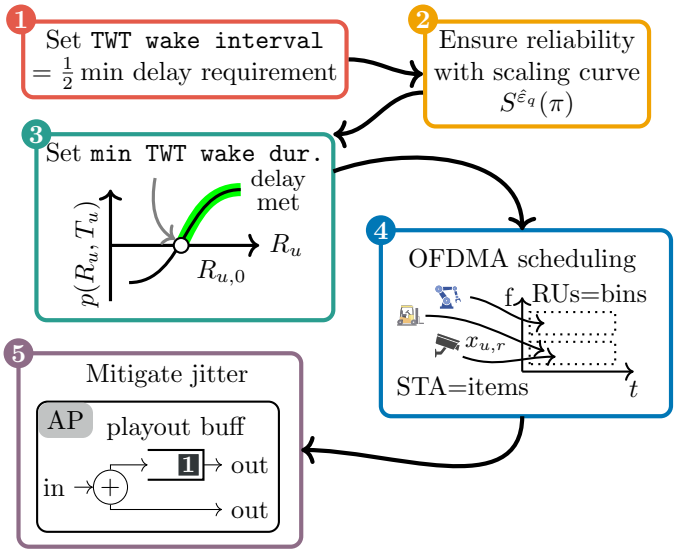

Fig. 6. PONTE step by step. (1) PONTE starts by setting the TWT period and (2) finds a scaling curve to ensure reliability. (3) It then finds the polynomial root to set an adequate rTWT session duration that meets delay requirements. Lastly, (4) it asserts the OFDMA scheduling and (5) when necessary, mitigates jitter with playout buffers.

and sets the same value for 802.1Qbv GCL gate opening time. In particular, PONTE selects $L_u, \forall u$ to meet the delay requirement Δ_u.

PONTE takes the stochastic scaling curve $S^{\hat{\varepsilon}_q}(\pi)$ of the previous step; sets the 1st rTWT session offset $\delta_u = T_u, \forall u$; and looks for a session duration L_u to satisfy inequality $h(\alpha_q^{(Tot)}, \beta_q^*) \leq \Delta_q$. From (25) we know $h(\alpha_q^{(Tot)}, \beta_q^*)$ is fully determined except for the rTWT rate for STA u, i.e. $R_u = \frac{C_r L_u}{L_u + T_u}$ with C_r the rate of each RU. Hence, PONTE looks for the smallest rate R_u to satisfy the delay requirement. It transforms the delay inequality into a polynomial $p(R_u, \delta_u)$ whose sign determines whether the delay requirement is met – see (40) and (41) in Appendix B.

Given $\delta_u = T_u$, we know $p(R_u, \delta_u)$ is a polynomial of degree $N+1$ on R_u. From empirical evidence, we observe that $p(R_u, T_u)$ is monotonically increasing for any value of $R_u \leq C_r$ satisfying the stability constraint (24), i.e. for any value in the interval

$$I = (I_0, I_1] = \left(\sum_{q' \prec q} C_{q'}^{(Tot)} + C_q \sum_{i=0}^{N} p^i,\ C_r \right] \tag{36}$$

Using Rolle's theorem [46], polynomial $p(R_u, T_u)$ has a single root $R_{u,0} \in I$, and PONTE runs Brent's method [47] to find it. Finally, it takes the computed root to obtain the `Minimum TWT wake duration`:

$$L_u = R_{u,0} \mathcal{T} / C_r. \tag{37}$$

In case the polynomial $p(R_u, T_u)$ is not finite, or has same sign at both extremes of the interval I (e.g. $p(I_0, T_u) < 0$ and $p(I_1, T_u) < 0$), it is not possible to meet the delay requirement (32). Hence, PONTE sets $L_u > \mathcal{T}$ to prevent STA u from being scheduled in the next step.

Step 4: *802.11 OFDMA scheduling.* PONTE assigns which RU r is used by each STA u in OFDMA transmissions – i.e., PONTE sets $x_{u,r}$. To decide the OFDMA scheduling $x_{u,r}$, it reduces Problem 1 to the GAP – as specified in the proof of Proposition 1. That is, PONTE creates a GAP instance with user-defined profits $p_{u,r}$; bins (RUs) of capacity 1; and item (STAs) weights $L_u/\mathcal{T}$ – with L_u obtained as detailed in (37).

To solve the GAP, PONTE leverages the FPTAS proposed in [48, Algorithm 2], a dynamic programming approach that sequentially solves the single knapsack problem [44] using the FPTAS Lawler algorithm [49]. Concretely, it invokes [48, Algorithm 2] repeatedly until no item (STA) assignment is found or all items are assigned. As a result, it specifies the OFDMA scheduling $x_{u,r}$ of every STA u.

Step 5: *jitter mitigation.* Lastly, PONTE checks whether the jitter requirement (33) is met. By construction and Corollary 4, PONTE election of $L_u, T_u, \delta_u, x_{u,r}$ fully determines the delay CDF $F_{d(t)}(x) = \mathbb{P}(d(t) \leq x)$. Consequently, it is possible to obtain the average delay [50, (A.9)] and its standard deviation j_q (jitter [11]) with the density function $f_{d(t)}(x) = \frac{d}{dx} F_{d(t)}(x)$. If the computed jitter j_q is stricter than the requirement J_q, PONTE passes the received traffic from queue q to a playout buffer [11, Chapter 9.2] that dispatches packets at a rate that matches the packet period, i.e. $\frac{l_{\max}^q}{C_q}$ seconds. PONTE meets the jitter requirement (32) with probability $1 - \varepsilon_q$ since the playout buffer guarantees[3] a zero delay variance between two packets with probability $1 - \varepsilon_q$.

Thanks to the GAP reduction from Step 4, PONTE provides the following guarantees.

Proposition 2. *Denote ϵ as the granularity of Lawler's algorithm, R as the number of RUs, and U the number of STAs to schedule. PONTE scheduler is an FPTAS algorithm with $(2+\epsilon)$-approximation guarantee, and computational complexity of $\mathcal{O}(R^2 U \log \frac{1}{\epsilon} + \frac{R^2}{\epsilon^4})$ worst case run time.*

Proof. PONTE complexity is governed by Step 4, in which it solves the OFDMA scheduling problem as the GAP. As specified in Step 4, PONTE repeatedly invokes [48, Algorithm 2] using Lawler [49] algorithm. Just a single invocation of [48, Algorithm 2] guarantees a $(2+\epsilon)$-approximation; and a worst-case run time $\mathcal{O}(RU \log \frac{1}{\epsilon} + \frac{R}{\epsilon^4})$ – see [48, Sec. I].

By construction of [48, Algorithm 2], its result ensures that the last bin (RU) has no more room for any item (STA). However, there may be room for more items in the other bins (RUs) e.g. if the dynamic programming approach may have moved an item from the first to the last bin. Hence, PONTE issues, at most, R invocations of [48, Algorithm 2] in case it exhausts the capacities of all bins (RUs). Consequently, the worst case run time is $\mathcal{O}(R^2 U \log \frac{1}{\epsilon} + \frac{R^2}{\epsilon^4})$.

Finally, PONTE guarantees reliability, delay and jitter requirements for time-sensitive traffic by construction of Step 2, 3 and 5 respectively. □

VIII. Results

We first validate the delay/reliability bounds (Sec. VIII-A), then we benchmark PONTE against a state of the art solu-

[3]It is possible to obtain the exact jitter j_q using Corollaries 3 and 4. However, we omit it for brevity given j_q has an order of magnitude of 10^{-18} at the exit of the playout buffer.

TABLE II
CONSIDERED IIOT SCENARIO [13], [37]–[40]

traffic	period	pkt size	delay	reliability	jiter	STAs
Robot	8 ms	50 B	8 ms	99.99%	2 ms	5
Vehicle	100 ms	100 B	20 ms	99.99%	5 ms	3
Interactive video	2 ms	1500 B	50 ms	99%	15 ms	2

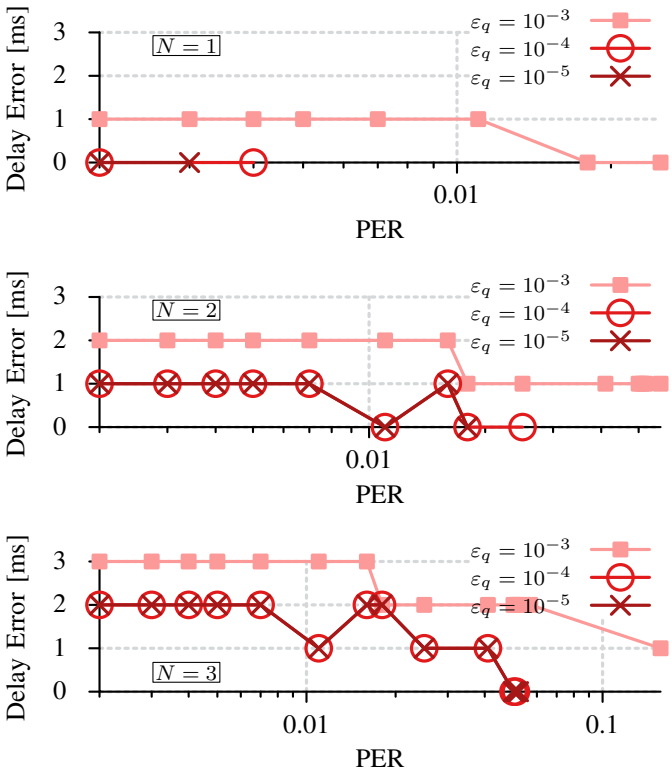

Fig. 7. Error of our delay bound $h(\alpha_q^{(Tot)}, \beta_q^*)$ for Robot traffic. We consider an STA with Robot and Interactive Video flows, different violation tolerance ε_q, PERs p and retransmissions: $N = 1$ (top), $N = 2$ (middle) and $N = 3$ (bottom).

tion through an extensive simulation campaign (Sec. VIII-B). The experiments consider an IIoT scenario with traffic flows of Robots, Vehicles and Interactive Video [13], [37]–[40]. Table II summarizes the traffic profiles of each traffic flow.

A. Validation of delay & reliability bounds

We consider an 802.11ax Wi-Fi setup with a single STA and AP. The STA has two queues governed by 802.1Qbv GCL: one for Robot traffic (high priority) and the other with Interactive Video traffic (low priority) – see TABLE II.

As considered by [34] and PONTE (Steps 1, 3), we assume rTWT sessions are aligned with 802.1Qbv GCL. We conduct OMNET++ Wi-Fi emulations using the INET framework implementation of 802.1Qbv with different PERs p. To derive the PERs we use MATLAB simulations of an 802.11ax channel [51] with 20 MHz bandwidth, a 3/4 coding rate, 64-QAM MCS, and diferent Bit Error Rates (BERs). The election of the small bandwidth and coding rate is motivated by the small bitrate – see Table II – and robustness considerations [52].

In the validation we consider increasing PERs p, retransmissions N, and the following violation tolerances $\varepsilon_q = 10^{-3}, 10^{-4}, 10^{-5}$ for the Robot traffic – see Table II. Note that these parameters are given by the scenario at hand, the quality of wireless link and the type of traffic; they are not configuration variables in PONTE. We then obtain the adequate stochastic scaling curve parameter $\hat{\varepsilon}_q$ – see (35) – and compute the delay bound $h(\alpha_q^{(Tot)}, \beta_q^*)$.

Fig. 7 shows the difference of our bound $h(\alpha_q^{(tot)}, \beta_q^*)$ with respect to the $1 - \varepsilon_q$ delay percentile from 100 OMNET++ emulations ($d_{\text{OPP},1-\varepsilon_q}$), each comprising 80 seconds of Robot traffic and Interactive Video flow. The error is computed as $h(\alpha_q^{(tot)}, \beta_q^*) - d_{\text{OPP},1-\varepsilon_q}$, hence positive errors means our bounds remain conservative. In the experiments we set the 802.1Qbv GCL – hence the corresponding rTWT parameters – to $L_u = 1$ ms and $T_u = 5$ ms. Lines with markers stop in Fig. 7 when the delay violation requirement (32) is not met for a specific PER.

In Fig. 7 we observe how the bound error increases with strict violation tolerances ε_q and number of retransmissions N. Nevertheless, more retransmissions increase the chance of a packet reaching the destination, thus of meeting the delay violation requirement (32) at higher PERs – see the proof of Corollary 4. In particular, in Fig. 7 (top) we observe that the delay violation requirement $\varepsilon_q = 10^{-3}$ is met with $N = 1$ retransmission for PERs above the 1%. Whilst with $N = 3$ retransmissions the delay violation requirement $\varepsilon_q = 10^{-3}$ is met for PERs above the 10% – see Fig. 7 (bottom).

Fig. 7 also shows that our bound is conservative (positive error), and remains smaller than 3 ms for a reasonable number of retransmissions $N \leq 3$. In brief, we effectively bound the maximum delay of Robot traffic with a 99.999% reliability.

B. Numerical results

We now assess PONTE's performance using a Python-based packet simulator[4] that mimics 802.1Qbv, rTWT, and 802.11ax OFDMA transmissions. As with the validation, we use a 20 MHz bandwidth, 3/4 coding rate, 64-QAM MCS, and up to $N = 2$ retransmissions. We consider OFDMA uplink transmissions and splits the available bandwidth into up to $R = 4$ RUs.

In the simulations, we consider the industrial scenario from TABLE II, which specifies the traffic characteristics and number of STAs of each traffic type. Unless specified, we run PONTE with granularity $\epsilon = 0.01$ and the profit function specified in (34), with $\theta = 0.02$ to slightly foster bandwidth demanding flows as Interactive video. The granularity is set to $\epsilon = 0.01$ for higher granularities took longer without improving the result. Lastly, as in the validation, simulations comprise a time lapse of 80 seconds.

Benchmarks. We compare PONTE against TASPER [19], a solution to schedule time-critical IIoT traffic using TWT. TASPER does not consider multi-user OFDMA nor 802.1Qbv queues. For the sake of a fair comparison we enhance TASPER to use multi-user OFDMA transmissions and manage multiple

[4]https://github.com/MartinPJorge/rtwtruscheduler

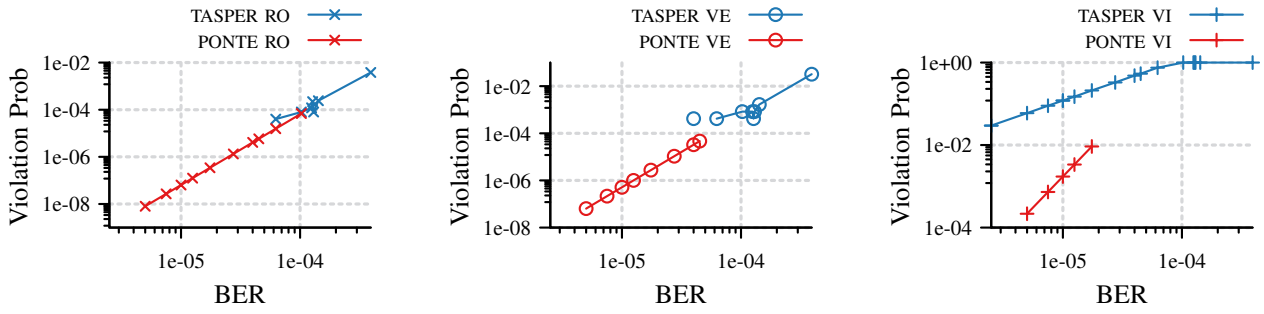

Fig. 8. Delay violation probability vs. BER for Robot (left), Vehicle (middle) and Inter. Video (right) traffic.

queues of a STA using 802.1Qbv. The enhanced version of TASPER repeats over and over the next steps. In *step 1* TASPER finds the RU that has been idle for longer. Then, in *step 2* TASPER computes the transmission rate of such RU and schedules a rTWT session for the STA that dispatches its oldest buffered packet faster. Lastly, in *step 3* TASPER opens the gates of all 802.1Qbv queues within the STA for which it scheduled the rTWT session. With these three steps TASPER benefits from concurrent OFDMA transmissions and 802.1Qbv priority scheduling. Thus, the enhanced version avoids packet collisions and minimizes the delay of time-critical traffic. Lastly, it is worth mentioning that the enhanced version of TASPER implementation also uses a neighborhood of 11 TWT sessions when looking for buffered packets in step 2. As in [19], packets buffered for more than 11 TWT session are discarded. We also compare PONTE against the optimal solution "OPT", which is devised through exhaustive search to solve Problem 1.

BER stress test. Obviously, the ideal channel has 0 BER, but it is almost impossible in practical scenarios. For that reason, in Fig. 8 we try out different BERs – representing channels with different qualities – in the 802.11ax channel conditions. The objective is to study the impact of the channel quality on PONTE and TASPER delay violation probability. Results show that PONTE does not exceed the maximum violation tolerance $\varepsilon_{\text{RO}} = \varepsilon_{\text{VI}} = 10^{-4}, \varepsilon_{\text{VI}} = 10^{-2}$ of each traffic type – see TABLE II reliabilities. Namely, Fig. 8 shows how PONTE stops scheduling traffic (line stops) in high BERs situations to not exceed the tolerable violation probabilities. Conversely, TASPER schedules traffic despite the high BERs because it does not take into account the channel conditions. Moreover, TASPER always exceeds the maximum violation requirement $\varepsilon_{\text{VI}} = 10^{-2}$ for Interactive Video traffic because it eagerly schedules traffic with tighter delay bounds (Robot, Vehicle).

STAs stress test. We now stress PONTE to schedule an increasing number of STAs. From now on, we use `1x` traffic multiplier to denote that the considered scenario has the number of STAs specified in TABLE II. Fig. 9 shows the impact of increasing the traffic multiplier in steps of `0.25x`.

In Fig. 9a we observe that higher granularities (smaller ϵ) achieve a larger normalized objective, matching the optimal solution. Above `2.5x` traffic multiplier, TASPER achieves a greater objective for it schedules more STAs than PONTE. However, TASPER schedules more STAs for it pitfalls into a 20%-25% of VI delay violations – see Fig. 9e.

TABLE III
SCHEDULED STAS PER PROFIT PARAM θ AND TRAFFIC MULTIPLIER nX

nx \ θ	-10^{-1}	-10^{-2}	-10^{-3}	0	10^{-3}	10^{-2}	10^{-1}
3x	26	26	26	26	27	27	**28**
4x	34	34	34	34	34	**35**	28
5x	41	41	41	41	**42**	36	28

The key of PONTE is its ability to perform *admission control*, traffic whose requirements cannot be met is rejected. Fig. 9d shows that PONTE decreases the admission of VE and VI packets for traffic multipliers above `2.5x` to prevent violating their time-sensitive requirements. In particular, PONTE rejects the admission of VE and VI STAs setting $x_{\text{VE},r} = x_{\text{VI},r} = 0, \forall r$. Such admission control applied to large traffic volumes, leads to a drop in the Jain's Fairness [53] as depicted in Fig. 9c.

In Fig. 9f we observe that PONTE ensures a near-zero jitter thanks to the use of the playout buffers. Moreover, Fig. 9b shows that PONTE runs two orders of magnitude faster than TASPER even with fine grained settings as $\epsilon = 0.001$. Moreover, PONTE is invoked only once to set the periodic rTWT schedules, whereas TASPER is invoked to schedule the traffic of every single TWT session. Hence, a single execution of 12 sec (a warm-up) is enough for PONTE to schedule a persistent periodic exchange between STAs and AP.

In Fig. 10 we observe that TASPER schedules more VI and VE STAs for `3x` and `5x` traffic multipliers, at the expense of violating (Fig. 9e) the delay reliability requirement of VI traffic. Fig. 11 provides more insights into the delay experienced by each traffic class with a traffic multiplier `1x`. We observe that the delay requirement (32) is always met with PONTE whereas TASPER violates it for VI traffic.

Profit impact. We recall the definition of the profit is a network administration decision. All results up to this point were optimal using a profit function (34) with $\theta = 0.02$. However, we remark that PONTE may schedule more STAs by setting θ appropriately. To this end, we carry out a sensibility analysis of the impact of θ on large traffic multipliers the results of which are shown in TABLE III. They suggest that an adequate setting of θ may lead to scheduling significantly more STAs (14 more STAs for the `5x` traffic multiplier). Given the results, we set the order of magnitude for the absolute value

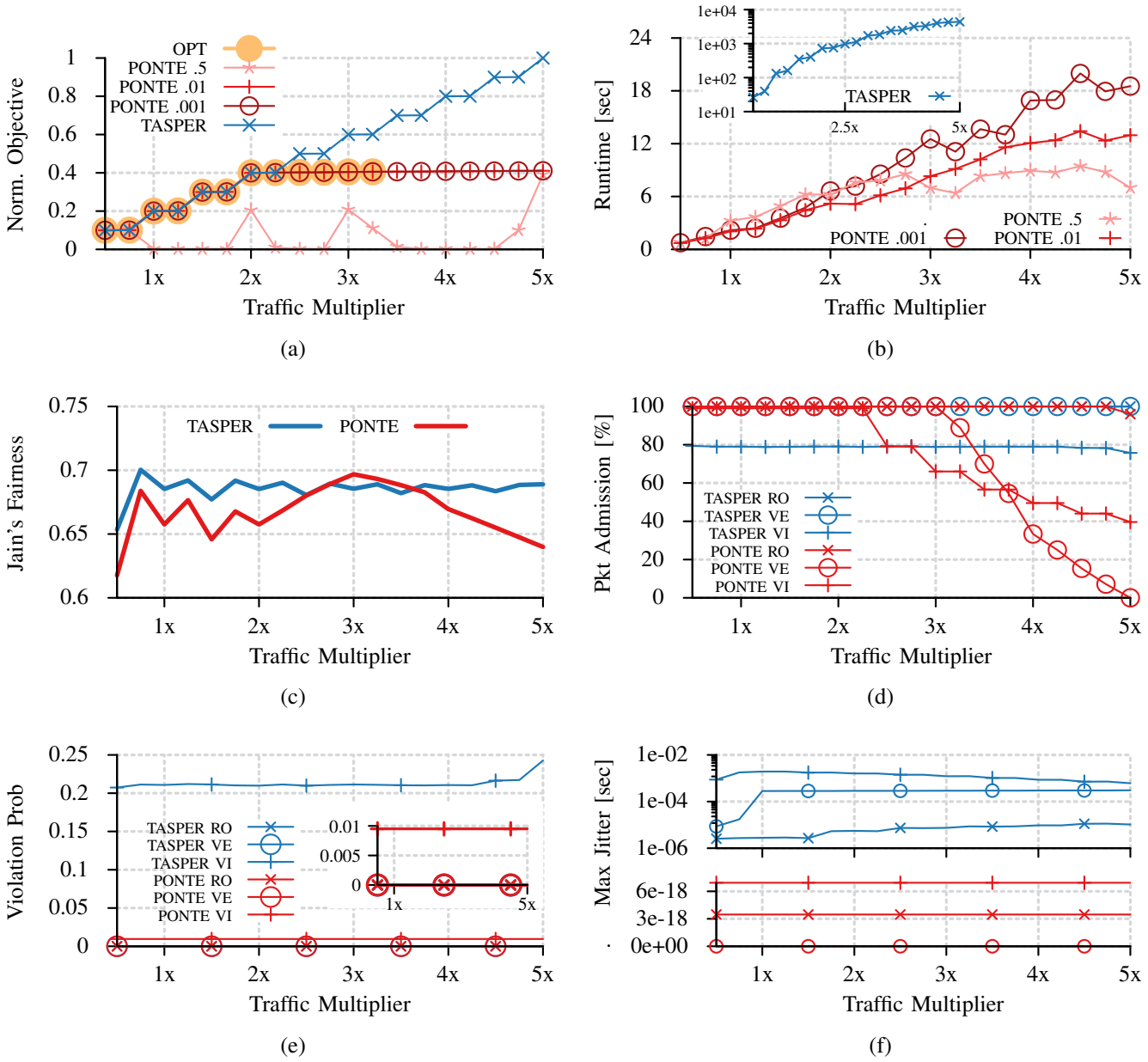

Fig. 9. Impact of the traffic multiplier on PONTE and TASPER performance. Fig. 9a and 9b show PONTE performance with different granularities ϵ. Fig. 9c to 9f show the impact of TASPER and PONTE ($\epsilon = 0.01$) scheduling decisions on Robot (RO) Vehicle (VE) and Interactive Video (VI) time-sensitive traffic.

of the profit parameter $|\theta|$ smaller than 10^{-3}. Such parameter election guarantees a stable admission regardless the traffic volume – see Table III.

IX. Implementation Limitations

To the best of our knowledge, `Openwifi` [54] is the only open-source solution to configure the Wi-Fi behaviour in commodity software defined radio hardware. At the time of writing this manuscript the open-source version of `Openwifi` does not offer rTWT, MU OFDMA nor 802.1Qbv as open-source features. Such features are implemented in the `openwifi+` subscription[5], specifically, the latter implements: (i) UL OFDMA with one user; (ii) a configurable OFDMA scheduler; and (iii) rTWT & enhanced EDCA. Moreover, there is another subscription offering a TSN evaluation kit[6] that implements (iv) 802.1Qbv scheduling.

We identify the following limitations/challenges to implement PONTE with `openwifi` subscriptions:

(L1) `openwifi+` supports UL OFDMA for just a single user, whereas PONTE performs RU assignment for multiple STAs.

(L2) It is not clear to which extent the OFDMA scheduler is configurable with `openwifi+` subscription, i.e. whether it is possible to implement a custom algorithm as PONTE or just to *configure* policies as e.g. proportional fairness.

[5]https://openwifi.tech/subscriptions/openwifi

[6]https://openwifi.tech/subscriptions/w-tsn-evaluation-kit

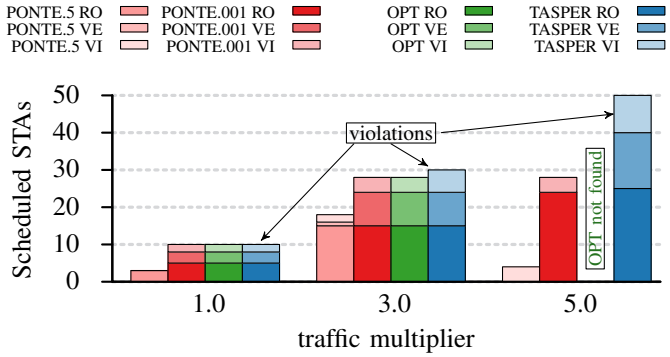

Fig. 10. Scheduled STAs vs the traffic multiplier for PONTE with granularities $\epsilon = 0.9, 0.001$, TASPER and optimal (OPT) solution.

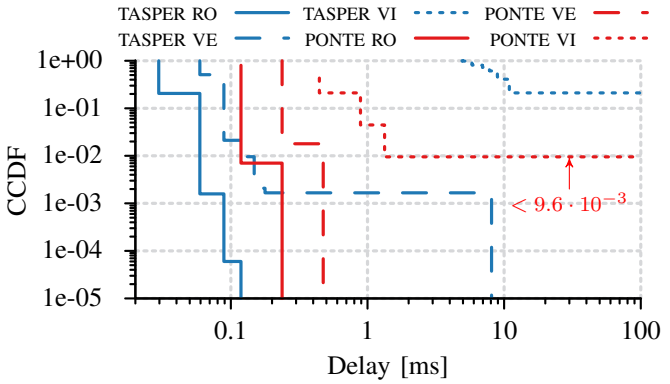

Fig. 11. Delay CCDF $\mathbb{P}(d(t) > x)$ of TASPER and PONTE for Robot (RO), Vehicle (VE) and Int. Video (VI) traffic with a traffic multiplier `1x`.

(L3) `openwifi+` does not specify whether it implements MU OFDMA transmissions within rTWT sessions.

(L4) `openwifi+` TSN evaluation kit does not specify to which extent it is possible to configure the 802.1Qbv scheduling, concretely if the GCL can be aligned with the rTWT sessions as PONTE does.

On top of the aforementioned limitations, PONTE performance guarantees hold in IIoT environments in which every STA implements rTWT, 802.1Qbv, and 802.11ax MU OFDMA. Heterogeneous IIoT environments with e.g., a STA that does not implement rTWT would result in potential collisions within the rTWT sessions scheduled by PONTE for other STAs thereby resulting in reliability/delay violations.

In summary, the open-source and subscription versions of `openwifi` are not yet ready to support PONTE. Additionally, PONTE is a solution that works in IIoT scenarios with every STA implementing, at least, rTWT.

X. Conclusions

This work provides worst case bounds with a degree of reliability for IEEE 802.11 networks implementing 802.1Qbv, OFDMA and rTWT, promising enablers for wireless TSN. Our analysis – carried out within the framework of Network Calculus – captures the unpredictable nature of wireless networks.

Consequently, we present PONTE, a FPTAS algorithm that leverages our analysis and schedules TSN transmissions in time and frequency using rTWT sessions and multi user OFDMA. PONTE also employs playout buffers to satisfy the tight jitter requirements of TSN traffic. We use OMNET++ to validate the analytical reliability bounds, and compare PONTE with a reference solution in a rigorous simulation campaign. Results show that PONTE is `100x` faster and meets the tight reliability constraints 99%, 99.99% which are significantly violated by the reference state-of-the-art solution.

In future work we plan to perform a thorough sensibility analysis of the ϵ, θ parameters of PONTE to: (i) provide an optimality-runtime trade-off; and (ii) understand how to adapt the profit θ to upon demand changes. We also plan to extend PONTE to decide when each 802.1Qbv queue opens its gate within the rTWT session. Thus, we will also compute the service curve offered to each queue when not all 802.1Qbv gate openings are alligned within the rTWT session – as assumed in Corollary 1.

Appendix A
Arrival Rate Bound

Note the delay bound in (1) goes to ∞ if the rate of the strict affine arrival curve $\gamma_{C,b}(t)$ is greater than the service rate, i.e. if $C > r$. To check whether $C > r$, we use the expression of the strict affine arrival curve of a queue with incoming retransmissions due to packet errors – see Corollary 2. In particular we get $C = \sum_{i=0}^{N} p^i C_q$, being p the packet loss probability; N the maximum number of retransmissions; and C_q the arrival rate of e.g. interactive video at the queue – which is $C_q = \frac{1\,\text{pkt}}{2\,\text{ms}} = 500\,\text{pps}$ according to TABLE II. For each interactive video packet of $50\,\text{B}$, in a scenario with no retransmissions ($N = 0$) we have $C < r$ as long as the service rate is $r > 6\,\text{Mbps}$. Hereof, the delay bound in (1) is finite. In case $C > r = 6$, the flow cannot be attended with delay guarantees and is not accepted by the solution we propose in Sec. VII.

Appendix B
Delay Polynomial

In this appendix we define a polynomial $p(R_u, \delta_u)$ whose sign determines whether the delay constraint (32) is met or not. The idea is to unroll the inequality $h(\alpha_q^{(Tot)}, \beta_q^*) \leq \Delta_q$.

First of all, we check what is the $\sum_{j=0}^{N} b_{q,j,\infty}$ term present in $h(\alpha_q^{(Tot)}, \beta_q^*)$ – see (25).

$$\sum_{j=0}^{N} b_{q,j,\infty} = \frac{1}{\det(A)} \sum_{j=0}^{N} p^j C_q \sum_{i=1}^{j} \det(A_i)$$

$$\boxed{+ b_q \frac{1-p^{N+1}}{1-p} + \frac{1-\hat{\varepsilon}}{1-p}\left(N - \frac{p(p^N-1)}{p-1}\right) + C_q W p \frac{Np^{N+1} - (N+1)p^N + 1}{(1-p)^2}} \; f_q(N)$$

Then, $h(\alpha_q^{(Tot)}, \beta_q^*) \leq \Delta_q$ unrolls – using (25) – as

$$\frac{1}{\det(A)} \sum_{j=0}^{N} p^j C_q \sum_{i=1}^{j} \det(A_i) + f_q(N) + R_u \delta_u + \max_{q'' \succ q} \{l_{\max}^{q''}\} + \sum_{q' \prec q} b_{q'}^{(Tot)} \leq \Delta_q (R_u - \sum_{q' \prec q} C_{q'}^{(Tot)}) \quad (38)$$

Finally, we define our polynomial as

$$p(R_u,\delta_u) = -\det(A)\Bigg\{\Delta_q(R_u - \sum_{q'\prec q} C_{q'}^{(Tot)}) - f_q(N) - R_u\delta_u - \max_{q''\succ q}\{l_{\max}^{q''}\} - \sum_{q'\prec q} b_{q'}^{(Tot)}\Bigg\} + \sum_{j=0}^{N} p^j C_q \sum_{i=1}^{j}\det(A_i) \tag{39}$$

Consequently, we ensure the delay constraint (32) is met if the following holds

$$p(R_u,\delta_u) \le 0, \quad \text{if}\det(A) \ge 0 \tag{40}$$

$$p(R_u,\delta_u) \ge 0, \quad \text{if}\det(A) < 0 \tag{41}$$

To speed up the computation of both $\det(A), \sum_j \det(A_j)$ we provide closed-form expressions for both in in Appendices C and D, respectively.

APPENDIX C
EXPRESSION FOR $\det(A)$

To calculate the determinant of matrix A from Corollary 2 one can rely on the properties of the determinants.

Subtracting the first row to the others will leave the determinant unchanged

$$\det(A) = \begin{vmatrix} d_1 & a_2 & a_3 & \dots & a_N \\ a_2 - d_1 & d_2 - a_2 & 0 & \dots & 0 \\ a_3 - d_1 & a_3 - a_2 & d_3 - a_3 & \ddots & 0 \\ \vdots & & & \ddots & \vdots \\ a_N - d_1 & a_N - a_2 & & \dots & d_N - a_N \end{vmatrix} \tag{42}$$

Now, it is possible to make zeros in the first row using multiples of the others. What remains is the determinant of a triangular matrix, that can be calculated by multiplying the elements of its diagonal. Hence,

$$\det(A) = \left(d_1 - \sum_{i=2}^{N}(a_i - d_1)K_i\right)\prod_{i=2}^{N}(d_i - a_i)$$

with the K_i term fully determined as follows

$$K_i = \begin{cases} \frac{a_i - \sum_{j=i+1}^{N}(a_j - a_i)K_j}{d_i - a_i}, & 2 \le i < N \\ \frac{a_N}{d_N - a_N}, & i = N \end{cases}$$

APPENDIX D
EXPRESSION FOR $\sum \det(A_j)$

Let $\{a_1, a_2, a_3, ..., a_{N-1}, a_N\}$ be the columns of A, where A is defined in Corollary 2. Then, $A_j = \{a_1, ..., a_{j-1}, \phi, a_{j+1}, ..., a_{N-1}, a_N\}$, for ϕ defined in Corollary 2.

Notice that

$$\det(A_1) + \det(A_2) = |\phi, a_2, ..., a_N| + |a_1, \phi, ..., a_N| = |\phi, a_2, ..., a_N| - |\phi, a_1, ..., a_N| = |\phi, a_2 - a_1, ..., a_N|,$$

and,

$$\det(A_1) + \det(A_2) + \det(A_3) = |\phi, a_2 - a_1, a_3, ..., a_N| + |a_1, a_2, \phi, ..., a_N| = |\phi, a_2 - a_1, a_3, ..., a_N| - |\phi, a_2, a_1, ..., a_N| = |\phi, a_2 - a_1, a_3, ..., a_N| - |\phi, a_2 - a_1, a_1, ..., a_N| = |\phi, a_2 - a_1, a_3 - a_1, ..., a_N|.$$

By induction, is easy to prove that

$$\sum_{j=1}^{N}\det(A_j) = |\phi, a_2 - a_1, a_3 - a_1, ..., a_{N-1} - a_1, a_N - a_1|$$

which is lower triangular but the first column. Knowing that a matrix and its transpose has the same determinant, the above expression is equivalent to (42), but substituting the first row by ϕ. Hence, the expression can be simplified as:

$$\det(A) = \left(\phi_1 - \sum_{i=2}^{N}(a_i - d_1)K_i'\right)\prod_{i=2}^{N}(d_i - a_i)$$

with the K_i' term fully determined as follows

$$K_i' = \begin{cases} \frac{\phi_i - \sum_{j=i+1}^{N}(a_j - a_i)K_j'}{d_i - a_i}, & 2 \le i < N \\ \frac{\phi_N}{d_N - a_N}, & i = N \end{cases}$$

REFERENCES

[1] N. E. Sanders, E. Şener, and K. B. Chen, "Robot-related injuries in the workplace: An analysis of osha severe injury reports," *Applied Ergonomics*, vol. 121, p. 104324, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003687024001017

[2] E. Matheson, R. Minto, E. G. G. Zampieri, M. Faccio, and G. Rosati, "Human–robot collaboration in manufacturing applications: A review," *Robotics*, vol. 8, no. 4, 2019. [Online]. Available: https://www.mdpi.com/2218-6581/8/4/100

[3] B. Sarkar and S. Bhuniya, "A sustainable flexible manufacturing–remanufacturing model with improved service and green investment under variable demand," *Expert Systems with Applications*, vol. 202, p. 117154, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422005462

[4] P. Avila-Campos, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Beacon-based wireless tsn association," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2022, pp. 1–2.

[5] S. Sudhakaran, V. Mageshkumar, A. Baxi, and D. Cavalcanti, "Enabling QoS for Collaborative Robotics Applications with Wireless TSN," in *2021 ICC Workshops*, 2021, pp. 1–6.

[6] O. Seijo, X. Iturbe, and I. Val, "Tackling the challenges of the integration of wired and wireless tsn with a technology proof-of-concept," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 7361–7372, 2022.

[7] A. M. Romanov, F. Gringoli, and A. Sikora, "A precise synchronization method for future wireless tsn networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3682–3692, 2021.

[8] P. M. de Sant Ana, N. Marchenko, B. Soret, and P. Popovski, "Goal-oriented wireless communication for a remotely controlled autonomous guided vehicle," *IEEE Wireless Communications Letters*, vol. 12, no. 4, pp. 605–609, 2023.

[9] H. Wang, J. Schmitt, and F. Ciucu, "Performance modelling and analysis of unreliable links with retransmissions using network calculus," in *Proceedings of the 2013 25th International Teletraffic Congress (ITC)*, 2013, pp. 1–9.

[10] J.-Y. Le Boudec and P. Thiran, *Network calculus: a theory of deterministic queuing systems for the internet*. Springer, 2001.

[11] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach (7th Edition)*, 7th ed. Pearson, 2017.

[12] J. Yan, W. Quan, X. Jiang, and Z. Sun, "Injection Time Planning: Making CQF Practical in Time-Sensitive Networking," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 616–625.

[13] R. Candell, K. Montgomery, M. Kashef Hany, S. Sudhakaran, and D. Cavalcanti, "Scheduling for Time-Critical Applications Utilizing TCP in Software-Based 802.1Qbv Wireless TSN," in *2023 IEEE 19th International Conference on Factory Communication Systems (WFCS)*, 2023, pp. 1–8.

[14] Z. Yang, Y. Zhao, F. Dang, X. He, J. Wu, H. Cao, Z. Wang, and Y. Liu, "CaaS: Enabling Control-as-a-Service for Time-Sensitive Networking," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 2023, pp. 1–10.

[15] X. He, X. Zhuge, F. Dang, W. Xu, and Z. Yang, "DeepScheduler: Enabling Flow-Aware Scheduling in Time-Sensitive Networking," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 2023, pp. 1–10.

[16] Q. Chen, Z. Weng, X. xu, and G. Chen, "A Target Wake Time Scheduling Scheme for Uplink Multiuser Transmission in IEEE 802.11ax-Based Next Generation WLANs," *IEEE Access*, vol. 7, pp. 158 207–158 222, 2019.

[17] Q. Chen and Y.-H. Zhu, "Scheduling Channel Access Based on Target Wake Time Mechanism in 802.11ax WLANs," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1529–1543, 2021.

[18] C. Yang, J. Lee, and S. Bahk, "Target Wake Time Scheduling Strategies for Uplink Transmission in IEEE 802.11ax Networks," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–6.

[19] C. Puligheddu, F. Busacca, R. Rusca, F. Raviglione, C. Casetti, C. F. Chiasserini, and S. Palazzo, "Target wake time scheduling for time-sensitive networking in the industrial iot," 2024, accepted in IEEE PIMRC 2024. [Online]. Available: https://hdl.handle.net/11583/2988104

[20] W. Qiu, G. Chen, K. N. Nguyen, A. Sehgal, P. Nayak, and J. Choi, "Category-based 802.11ax target wake time solution," *IEEE Access*, vol. 9, pp. 100 154–100 172, 2021.

[21] Q. Chen, "An Energy-Efficient Channel Access With Target Wake Time Scheduling for Overlapping 802.11ax Basic Service Sets," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18 973–18 986, 2022.

[22] D. Bankov, A. Didenko, E. Khorov, and A. Lyakhov, "OFDMA Uplink Scheduling in IEEE 802.11ax Networks," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[23] S. Bhattarai, G. Naik, and J.-M. J. Park, "Uplink Resource Allocation in IEEE 802.11ax," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[24] M. S. Kuran, A. Dilmac, O. Topal, B. Yamansavascilar, S. Avallone, and T. Tugcu, "Throughput-maximizing OFDMA Scheduler for IEEE 802.11ax Networks," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1–7.

[25] V. Gokhale, K. Kroep, R. V. Prasad, B. Bellalta, and F. Dressler, "ViTaLS—A Novel Link-Layer Scheduling Framework for Tactile Internet Over Wi-Fi," *IEEE Internet of Things Journal*, vol. 10, no. 11, pp. 9917–9927, 2023.

[26] B. Schneider, R. C. Sofia, and M. Kovatsch, "A Proposal for Time-Aware Scheduling in Wireless Industrial IoT Environments," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, 2022, pp. 1–6.

[27] Y. Jiang, Y. Liu *et al.*, *Stochastic network calculus*. Springer, 2008, vol. 1.

[28] O. Adamuz-Hinojosa, L. Zanzi, V. Sciancalepore, A. Garcia-Saavedra, and X. Costa-Pérez, "ORANUS: Latency-tailored Orchestration via Stochastic Network Calculus in 6G O-RAN," in *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*, 2024. [Online]. Available: https://arxiv.org/abs/2401.03812

[29] O. Adamuz-Hinojosa, V. Sciancalepore, P. Ameigeiras, J. M. Lopez-Soler, and X. Costa-Pérez, "A Stochastic Network Calculus (SNC)-Based Model for Planning B5G uRLLC RAN Slices," *IEEE Transactions on Wireless Communications*, vol. 22, no. 2, pp. 1250–1265, 2023.

[30] K. Xiong, S. Leng, C. Huang, C. Yuen, and Y. L. Guan, "Intelligent Task Offloading for Heterogeneous V2X Communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2226–2238, 2021.

[31] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment: Enhancements for Extremely High Throughput (EHT)," *IEEE P802.11be/D6.0*, pp. 1–1075, 2024.

[32] D. V. Bankov, A. I. Lyakhov, E. A. Stepanova, and E. M. Khorov, "Performance Evaluation of Wi-Fi 7 Networks with Restricted Target Wake Time," *Problems of Information Transmission*, vol. 60, no. 3, pp. 233–254, 2024.

[33] IEEE Standards 802.1, Institute of Electrical and Electronics Engineers, "Time-Sensitive Networking Task Group," 2024.

[34] D. Cavalcanti, C. Cordeiro, M. Smith, and A. Regev, "Wifi tsn: Enabling deterministic wireless connectivity over 802.11," *IEEE Communications Standards Magazine*, vol. 6, no. 4, pp. 22–29, 2022.

[35] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN," *IEEE Std 802.11ax-2021 (Amendment to IEEE Std 802.11-2020)*, pp. 1–767, 2021.

[36] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on ieee 802.11ax high efficiency wlans," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 197–216, 2019.

[37] F. Dressler, F. Klingler, M. Segata, and R. L. Cigno, "Cooperative driving and the tactile internet," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 436–446, 2019.

[38] D. Akhmetov, D. Das, D. Cavalcanti, J. Ramirez-Perez, and L. Cariou, "Scheduled time-sensitive transmission opportunities over wi-fi," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 1807–1812.

[39] K. Meng, A. Jones, D. Cavalcanti, K. Iyer, C. Ji, K. Sakoda, A. Kishida, F. Tsu, J. Yee, L. Li, and E. Khorov, "RTA report draft," 2018, iEEE 802.11-18/2009r6. [Online]. Available: https://mentor.ieee.org/802.11/dcn/18/11-18-2009-06-0rta-rta-report-draft.docx

[40] Industrial Internet Consortium, "Time Sensitive Networks for Flexible Manufacturing Testbed Characterization and Mapping of Converged Traffic Types," 2019. [Online]. Available: https://www.iiconsortium.org/pdf/IIC_TSN_Testbed_Char_Mapping_of_Converged_Traffic_Types_Whitepaper_20180328.pdf

[41] L. Zhao, P. Pop, and S. S. Craciunas, "Worst-case latency analysis for IEEE 802.1 Qbv time sensitive networks using network calculus," *Ieee Access*, vol. 6, pp. 41 803–41 815, 2018.

[42] C. Barroso-Fernández, J. Martín-Pérez, C. Ayimba, and A. De La Oliva, "Aligning rTWT with 802.1Qbv: A Network Calculus Approach," in *MobiHoc '23*. New York, NY, USA: Association for Computing Machinery, 2023, p. 352–354. [Online]. Available: https://doi.org/10.1145/3565287.3617606

[43] F. Ciucu, J. Schmitt, and H. Wang, "On expressing networks with flow transformations in convolution-form," in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 1979–1987.

[44] Kellerer, Hans and Pferschy, Ulrich and Pisinger, David and Kellerer, Hans and Pferschy, Ulrich and Pisinger, David, *Multidimensional knapsack problems*. Springer, 2004.

[45] T. Öncan, "A survey of the generalized assignment problem and its applications," *INFOR: Information Systems and Operational Research*, vol. 45, no. 3, pp. 123–141, 2007.

[46] M. Spivak, *Calculus*. Cambridge University Press, 2006.

[47] R. P. Brent, *Algorithms for minimization without derivatives*. Courier Corporation, 2013.

[48] R. Cohen, L. Katzir, and D. Raz, "An efficient approximation for the Generalized Assignment Problem," *Information Processing Letters*, vol. 100, no. 4, pp. 162–166, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020019006001931

[49] E. L. Lawler, "Fast Approximation Algorithms for Knapsack Problems," *Mathematics of Operations Research*, vol. 4, no. 4, pp. 339–356, 1979. [Online]. Available: http://www.jstor.org/stable/3689221

[50] H. C. Tijms, *Stochastic Models, an Algorithmic Approach*. John Wiley & Sons, 1994.

[51] Liu, Jianhan and Porat, Ron and Jindal Nihar, "IEEE 802.11ax Channel Model Document," IEEE, Tech. Rep. 802.11-14/0882r4, 09 2014.

[52] M. K. Atiq, R. Muzaffar, O. Seijo, I. Val, and H.-P. Bernhard, "When ieee 802.11 and 5g meet time-sensitive networking," *IEEE Open Journal of the Industrial Electronics Society*, vol. 3, pp. 14–36, 2022.

[53] R. K. Jain, D.-M. W. Chiu, W. R. Hawe *et al.*, "A quantitative measure of fairness and discrimination," *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, vol. 21, p. 1, 1984.

[54] 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.