Novel Radio Frame Design for Efficient Integration of Wireless Links into Time-Sensitive Networks

Youssef El Kaisi, Javier Villares, and Olga Muñoz
Dept. of Signal Theory and Communications
Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
E-mail: {youssef.el.kaisi, javier.villares, olga.munoz}@upc.edu

***Abstract*—This paper presents a novel radio frame design for wireless links that allows efficient scheduling of multiple time-sensitive flows with bounded delay and arbitrarily small packet jitter requirements. The proposed design is compatible with the IEEE 802.1 standards for Time Sensitive Networks (TSN) and permits the efficient integration of wireless links into newly arrived wired TSN infrastructure. With this aim, we have reformulated the window reservation mechanism introduced in standard 802.1Qbv to deal with the variable transmission rate of wireless links. The new radio frame is organized in a sequence of partially-overlapped windows. If the intended flow does not use its window completely, the unconsumed time can be scheduled to other flows in overlapped windows. The benefits of this approach are evaluated numerically in an illustrative industrial scenario showing substantial gain in terms of both reduced control cycle (flows' periodicity) and enhanced transmission throughput (number of admitted flows).**

***Index Terms*—Time-Sensitive Networking (TSN), radio resource management, scheduling, low-latency communications, packet jitter.**

I. Introduction

The automation of many industries demands the development of deterministic communication networks that allow the exchange of critical data with bounded and predictable latency. With this goal in mind, the IEEE 802.1 Time-Sensitive Networking (TSN) Task Group has developed a set of standards that enable the deployment of bridged Ethernet networks with bounded delay [1]. The adopted premise is to minimize buffering in the network bridges by reserving transmission windows at the bridge egress ports. This allows an incoming flow arriving at an ingress port to be immediately written to the destination egress port without waiting in a queue. The architecture relies on two standards: 802.1as [2], which enables precise synchronization of all network nodes, and 802.1Qbv [3], which defines the method for performing windowing.

Recently, there has been a significant interest in integrating wireless connections into emerging TSN wired networks. Various standardization groups and industrial consortia are working on new mechanisms to offer delay guarantees through wireless links [4] [5]. The primary obstacle to this goal is the inherent randomness of the wireless channel. While wired links can easily sustain stable high transmission rates, the transmission rate fluctuates randomly when connecting two nodes over the air due to channel fading and time-varying pathloss.

Although determinism is not strictly possible with wireless links, a clever design of radio frames allows for bounded latency with high probability. This means that, we have to accept inevitably that, with non-zero probability, some packets will not be decoded correctly or *in time* and will be declared lost. Under this premise, the straightforward design to minimize packet loss is to reserve a separate window in the radio frame for every flow and adjust its duration to provide sufficient time for transmitting the incoming packets at the lowest available data rate. While simple, this conservative design is inefficient because windows are not fully occupied when the channel realization is favorable, and the transmitter is allowed to select faster modulation and coding schemes (MCS).

To improve efficiency, in this article we study the design of radio frames with overlapped windows, avoiding spare time and, thereby, substantially increasing the system throughput. The design is focused on the downlink of a generic wireless link, although the approach is extensible to the uplink. The proposed design is assessed numerically in the simulation section for a wireless local area network operating in a typical industrial environment.

II. Problem Statement

We consider that K wireless terminals, attached to the same access point (AP), are connected simultaneously to a server from which they download individual data *flows*. In industrial applications, the server is typically a programmable logic controller (PLC) that periodically sends instructions to a set of K mobile machines or robots having wireless connectivity. These instructions are encapsulated into layer 2 frames, referred to as *packets* hereinafter, with a length of B_k bits and a periodicity given by the control cycle, T_k slots, for $k = 1, \ldots, K$. This type of traffic is commonly known as *isochronous* in the literature [6]. It is worth noting that time is assumed to be slotted so that all time intervals are multiple of the slot time T_{slot}, defined as the minimum scheduling unit.

Work supported by projects MAYTE (PID2022-136512OB-C21) and 6-SENSES (PID2022-138648OB-I00) funded by MCIN/AEI/10.13039/501100011033 and "ERDF A way of making Europe", EU, PREDICT-6G (grant 101095890) by EU's Horizon Europe, TIMING (TSI-063000-2021-145) by MINECO UNICO5G and, grants 2021 SGR 01033 and 2021 SGR 00772 by the Catalan Government (AGAUR)

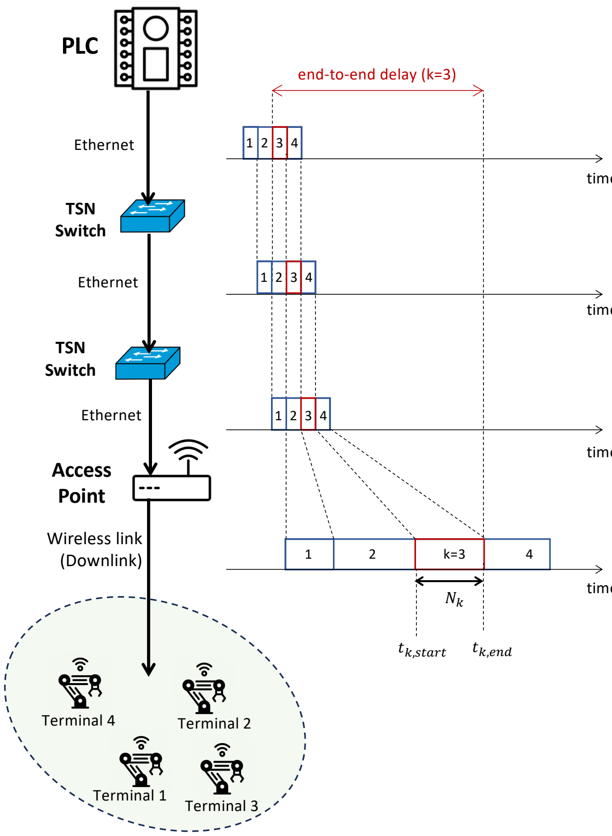

Fig. 1. End-to-end delay example. A window in the egress port of a node cannot be opened until the associated window in the ingress port is closed.

Our objective is twofold. Firstly, packets should be transferred from the PLC to the destination terminals with the minimum end-to-end delay. Secondly, the end-to-end delay should remain as stable as possible throughout the connection, thereby minimizing packet delay variation (jitter). The end-to-end delay of a packet is defined as the sum of the delays incurred at every link in the route from the PLC to the destination terminal. Following the 802.1Qbv recommendation, we assume that a transmission window opens in the egress port of every bridge as soon as the packet enters the bridge. This approach ensures that waiting times in queues become negligible and the end-to-end delay is primarily determined by the transmission time. However, while transmission time is deterministic and small in wired links, it tends to be variable and often longer in wireless links due to the randomness of the propagation channel.

Fig. 1 illustrates a simplified scenario with $K = 4$ flows following the same route from the PLC to the wireless access point. As depicted in Fig. 1, the contribution of the wireless link to the end-to-end delay is usually dominant. Consequently, it is necessary to design an efficient radio frame to transport the maximum number of isochronous flows from the AP to the terminals with bounded delay and predefined control cycles $\{T_k\}$.

The structure of the radio frame is as depicted in Fig. 2. A temporal window is assigned to every flow in the downlink. The kth window starts at time $t_{k,start}$, has a duration of N_k slots, and ends at time $t_{k,end}$. In Fig. 2, windows are independent (not overlapped). However, to multiplex more flows and improve efficiency, we will allow windows to overlap later, i.e., $t_{k+1,start} < t_{k,start} + N_k$. The objective is to design the optimum values for $t_{k,start}$ and N_k for $k = 1, \ldots, K$ and implement a dynamic scheduling algorithm to multiplex in time (Time-Division Multiple Access) the active K flows using the predefined windows as a reference. The role of windows in the scheduler procedure will be further explained in Sec. IV.

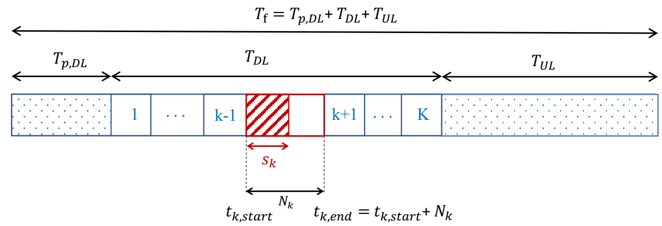

Fig. 2. Radio frame structure. The frame preamble ($T_{p,UL}$ slots) and the uplink section (T_{UL} slots) are marked using a blued dotted pattern. The downlink is divided into K independent windows with K the number of active flows. The burst transmitted in the kth window, with duration s_k slots, is indicated using a red striped pattern.

As highlighted in Fig. 2, the time required to transmit the packet of flow k is

$$s_k = B_k / R_k, \tag{1}$$

which varies over time because it depends on the time-varying transmission rate R_k, given in bits per slot. To compensate for this variability and avoid packet jitter, terminals can retain the received packet until the end of the window, $t_{k,end}$, before using it. This de-jittering procedure is a well-known concept in the literature and is often referred to as *hold-and-forward* in the context of 5G [8].

The (radio) frame duration T_f is adjusted to the greatest common divisor (GCD) of all the flow periods $\{T_k\}$. To simplify the analysis, we consider from now on that all the flows have the same period ($T_k = T$) and, therefore, the frame length is set to this value: $T_f = T$.

III. Non-overlapped windows design

In this section, we consider a simple frame design that reserves an exclusive window for each flow, with no overlap among windows, so that the transmission of the flow's packets is not affected by other flows. Typically, the transmission rate in a wireless link is selected according to the Signal-to-Noise-Ratio (SNR) at the receiver to ensure that the error probability due to transmission errors is negligible. The SNR is reported periodically to the transmitter and strongly depends on the propagation channel. As the channel is random, so are the transmission rate and time needed to send each packet, even for packets corresponding to the same flow. Consequently, the duration of each window shall be defined statistically.

The delay of packets in the wireless link implementing non-overlapped windows is directly proportional to s_k, thus, it is also random. Neglecting the propagation time, the delay is expressed as

$$d_k = \left(t_{k,start} - t_{k,end}^{wired}\right) + s_k. \quad (2)$$

The first term in (2) is the time between the closing instant of the window from the previous wired node, $t_{k,end}^{wired}$, and the instant at which the window in the wireless link opens, $t_{k,start}$. These instants are fixed and determined by how the network arranges the windows along the connection path. We consider that the network prioritizes the design of the wireless link, as it is the critical one, and then organizes the previous links accordingly so that $t_{k,end}^{wired} = t_{k,start}$ and hence the delay only depends on the second term, $d_k = s_k$.

The window of each flow has a limited duration, $N_k = t_{k,end} - t_{k,start}$. Packets with $s_k \leq N_k$ will generally experience different delays, which would cause jitter. To avoid it, the receiver can make use of the previously mentioned *hold-and-forward* mechanism [8] to increase the delay of all received packets so that they match the window duration N_k. For somewhat jitter-tolerant applications, it is also possible to match the delays to a value *lower* than N_k and allow some jitter at reception. The decision is up to the receiver and does not affect the frame design whatsoever.

However, there will also be packets with transmission times larger than the window duration, $s_k > N_k$, which will be lost because their delay is considered excessive. Thus, flows will suffer a non-null packet loss rate p_k. We expect user applications to have a certain tolerance to packet loss $\breve{p}_k$, such that, as long as the $p_k \leq \breve{p}_k$ the application will perform as expected:

$$p_k = \text{prob}\left(s_k > N_k\right) \leq \breve{p}_k. \quad (3)$$

The window duration N_k must be large enough to fulfil (3). Indeed, the larger N_k, the lower p_k, but also the longer the frame. Thus, N_k is selected as the smallest value that obeys (3), which in principle is the one that achieves the equality.

In practice, there is a finite number of available transmission rates R_k (and consequently transmission times s_k), so the possible values for the window duration N_k are limited to these options, which will in general not include the solution that reaches the equality in (3). Bearing this in mind, the window duration has to be obtained as follows:

$$N_k = \underset{N}{\text{argmax}} \; \text{prob}\left(s_k > N\right), \text{ s.t. } p_k \leq \breve{p}_k. \quad (4)$$

Note that the resulting packet loss probability will be, in general, lower than the required value ($p_k < \breve{p}_k$).

IV. Overlapped windows design

The frame design in Section III consists of contiguous, non-overlapped windows. The window length depends on the corresponding flow's channel statistics. Due to channel randomness, most transmitted packets will in general not use the whole window, leaving some slots empty. In this section, we present an alternative frame design that makes use of these empty slots to reduce radio frame length. We achieve this by designing windows that open before the previous ones close.

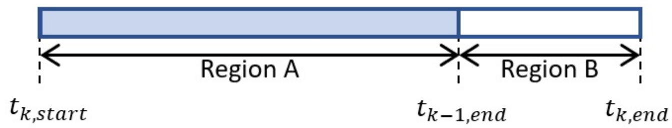

Fig. 3. Window configuration for the overlapped windows design

Fig. 3 shows the general structure of the window designed for the kth flow. We will refer to this window as the kth window henceforth. The window is composed of two regions. The first region, Region A, goes from the opening of the window $t_{k,start}$ until the closing of the previous window $t_{k-1,end}$. The second region, Region B, is the portion of the window beyond $t_{k-1,end}$ and until $t_{k,end}$. Although more than one window could close after $t_{k,start}$, to keep the explanation simple, we will assume that $t_{k-2,end} \leq t_{k,start}$.

In Region A, flow $k-1$ occupies as many slots as it requires during each packet transmission, and flow k is then allowed to make use of the leftover slots in the region. This reduces the amount of empty slots in window $k-1$ while not disturbing the operation of the previous flow, which is unaware of the overlap. In Region B, the kth flow does not overlap with previous windows and thus can use the resources without restrictions. The following window $k+1$ will overlap with window k, but this overlap will not affect the scheduling of flow k.

The occupancy of slots in region A by previous flows is random, characterized by a random variable M_k. Its distribution is parameterized by the opening instant $t_{k,start}$. The actual value of M_k affects the delay of the kth flow's packets in the following manner:

$$d_k = s_k + M_k. \quad (5)$$

The packet loss rate condition is thus given by:

$$p_k = \text{prob}\left(s_k + M_k > N_k\right) \leq \breve{p}_k. \quad (6)$$

As $t_{k,start}$ decreases, Region A becomes larger and Region B tends to decrease, reducing overall frame length. Indeed, the frame is at its minimum when $t_{k,start} = t_{1,start}$. We refer to this configuration as *full overlap*, since each window overlaps completely with all the previous ones. With this setting, window lengths $\{N_k\}$ increase with k, and the window associated to the last flow $k = K$ is as long as the whole frame. This makes this configuration generally not practical, since it results in flows with very different delays, most of them being intolerably large. To keep the delays under acceptable values, we impose a second condition when designing the windows: $N_k \leq \breve{d}_k$, where $\breve{d}_k$ is the maximum tolerable delay by flow k. Note that $\breve{d}_k \geq N_k^{min}$, where N_k^{min} is the non-overlapped window duration of flow k (window duration if $M_k = 0$). The particular case where $\breve{d}_k = N_k^{min}$ is referred to as the *minimum delay* configuration.

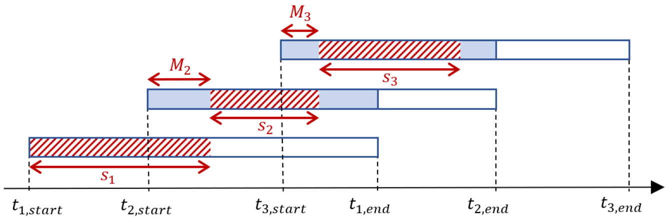

Fig. 4. Frame design example with 3 flows, highlighting the overlapped region of each window and showcasing the s_k slots used by the k-th flow for transmitting its packet.

The design of each window requires knowing the lengths and opening instants of previous windows. Thus, it is necessary to design windows recursively in ascending order as given by their indexation k. Note that flow 1 does not have any previous windows and thus is designed as if it was non-overlapped ($N_1 = N_1^{min}$). In practice, the following iterative procedure is used to design the kth window with as much overlap as allowed by the design constraints:

$t_{k,start} \leftarrow t_{k-1,end} - \breve{d}_k$
if $t_{k,start} < 0$ **then**
    $t_{k,start} \leftarrow 0$
end if
$N_k \leftarrow \infty$
while $N_k > \breve{d}_k$ **do**
    $t_{k,start} \leftarrow t_{k,start} + 1$
    $F(x) \leftarrow CDF(s_k + M_k)$
    $N_k \leftarrow min(x) : F(x) \geq 1 - \breve{p}_k$
end while

In Fig. 4 we provide a graphical example of the overlapped design. Overall, this design presents an intrinsic trade-off between delay and frame length, where the extremes are the *minimum delay* configuration ($\breve{d}_k = N_k^{min}$), resulting in a long radio frame, and the *full overlap* configuration, yielding minimum frame length but large delays. It is necessary to highlight that even the minimum delay N_k^{min} is already substantially larger than the delay in the wired links, so it is important to weigh the improvement on frame length versus delay increase when designing the frame.

V. Simulation results

This section provides a quantitative evaluation of the proposed frame design through software simulation. The simulations portray the performance of the design in terms of control cycle (frame duration) and throughput as the number of active flows grows.

1) Wireless network: A wireless access network founded on the standard 802.11ax [7] is considered in this section for evaluating the proposed windows design. To avoid the random delay in legacy WiFi (contention-based access), we adopt the standardized Orthogonal Frequency Division Multiple Access (OFDMA) [9], mirroring the medium access in 5G networks. The time slot duration is set to $T_{slot} = 13.6\mu s$, which corresponds to one OFDM symbol with a 800 ns guard interval. The transmission bandwidth is set to BW = 20 MHz and is divided into 256 subcarriers spaced 78.125 kHz, 234 of which are used for data transmission. For simplicity, in every slot, the scheduler is compelled to assign all the data subcarriers to the same flow, hence multiplexing flows solely in time.

For the transmission of each packet, the MCS is selected based on the *effective SNR*, a link quality indicator that is a function of the SNR of all the data subcarriers. From the MCS table in the standard, comprising 11 schemes, we select the highest MCS that guarantees a Block Error Rate (BLER) not higher than 10^{-4} for a LDPC-encoded block of 1458 bytes. A low BLER is adopted because retransmissions are not feasible for the studied delay-constrained system. The effective SNR is evaluated using the Received Bit Information Rate (RBIR), a mutual information measure that was adopted by the 802.11ax task group for physical-layer abstraction [10] [11].

2) Wireless channel: We consider a random fading channel with exponentially-decaying power-delay profile and delay spread $\tau = 50$ ns [12]. Assuming Rayleigh fading, the discrete channel impulse response is given by

$$h[n] = Ce^{\frac{-nT_s}{2\tau}} a[n], \tag{7}$$

where $\{a[n]\}_{n\geq 0} \sim \mathcal{CN}(0, 2)$ are complex-valued zero-mean Gaussian-distributed random variables of unitary (per-component) variance, C a normalization constant fixing $\sum_{n=0}^{\infty} \mathbb{E}\{|h[n]|^2\} = 1$, and $T_s = 1/BW$ the sampling period. We assume also uncorrelated scattering, i.e., $\mathbb{E}\{h[n_1]h^*[n_2]\} = 0$ for any $n_1 \neq n_2$.

The above delay spread ($\tau = 50$ ns) results in a coherence bandwidth of around 1.6 MHz, defined as the frequency separation for which the autocorrelation of the channel frequency response $H(f) = \sum_{n=-\infty}^{\infty} h[n]e^{-j2\pi f n T_s}$ decreases by a factor of 0.9. Furthermore, the coherence time is assumed to be longer than the control cycle (frame length), so the resulting fading channel is flat in time, but frequency selective.

The SNR of a given subcarrier at frequency f is expressed as follows:

$$SNR(f) = \gamma \cdot |H(f)|^2, \tag{8}$$

with $\mathbb{E}\{|H(f)|^2\} = \sum_{n=0}^{\infty} \mathbb{E}\{|h[n]|^2\} = 1$ and γ the average received SNR per subcarrier.

3) Traffic: For the sake of simplicity and clarity, in the simulations we configure the flows to be identical, i.e., they share the same periodicity (T_f), the associated channels are independent and identically distributed, and they all share the following parameters:

- $\gamma = 20$ dB
- $B = 1500$ Bytes
- $\breve{p} = 2 \cdot 10^{-3}$

A. Control cycle simulation

The control cycle is determined by the frame duration T_f. As shown in Fig. 2, the total frame length depends on the downlink and uplink portions and the frame overhead. In the simulation, we compute the time required to allocate the designed downlink windows in the downlink section and

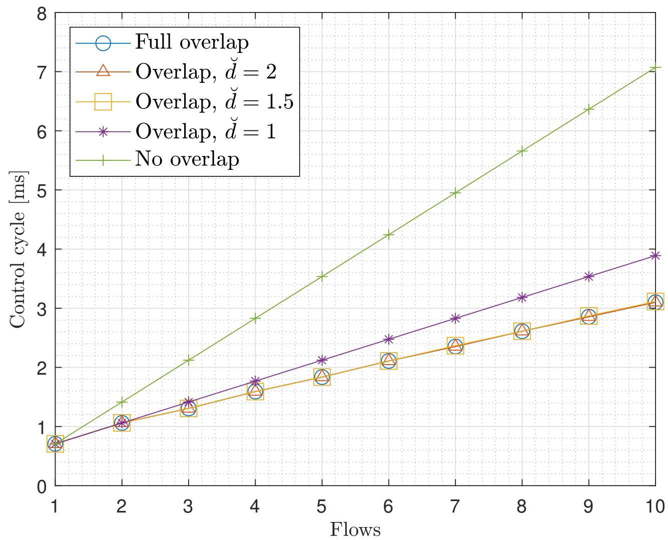

Fig. 5. Control cycle for different delay constraints. Delay constraints are normalized by the non-overlapped window length $N^{min} = 52$ slots.

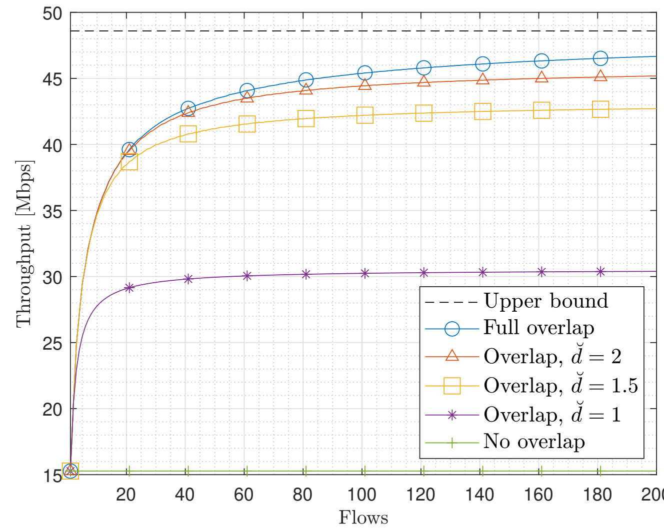

Fig. 6. Supported throughput for different delay constraints. Delay constraints are normalized by the non-overlapped window length $N^{min} = 52$ slots.

assume that the uplink part will be of equal length, i.e., $T_{UL} = T_{DL}$. This assumption corresponds to a scenario with balanced traffic load in both directions. Accordingly, the frame duration is given by

$$T_f = T_{p,DL} + T_{DL} + T_{UL} = \frac{2}{0.9} T_{DL}. \quad (9)$$

In this equation, we are also assuming that the overhead $T_{p,DL}$ approximately accounts for a 10% of the frame length T_f (factor 1 over 0.9 in (9)), which is a widely-accepted overhead in radio packet communications.

Fig. 5 shows the minimum allowed control cycle as the number of flows increases for the non-overlapped design and the overlap design case with different delay constraints $\breve{d}$. The non-overlapped window duration N^{min} is equal to 52 slots (707.2 μs), which is the transmission time associated to MCS1 (QPSK modulation and code rate 1/2). This MCS is selected by solving (4). The delay constraints have been selected based on N^{min} and expressed normalized by this value in the figure. Specifically, the chosen delay constraints are 1 (minimum delay), 1.5, and 2 times N^{min}, respectively.

All configurations show a linear increase with the number of flows, although with different slopes. For constraints $\breve{d} = 2$ and $\breve{d} = 1.5$, their curves coincide with each other and barely differ from the full overlap case. More interesting is that the minimum delay constraint, $\breve{d} = 1$, even though it results in a notable increase in control cycle with respect to the full overlap case, still provides a much shorter control cycle than the non-overlapped windows design, all while constraining the delay to the same minimum value.

B. Throughput simulation

The overall throughput is the amount of information transmitted per unit of time. In this simulation we evaluate the throughput that the configurations in Fig. 5 provide. The throughput is estimated as:

$$TH = \frac{2K \cdot B}{T_f} = 0.9 \frac{K \cdot B}{T_{DL}}, \quad (10)$$

with K being the number of downlink flows. As in (9), we assume that $T_{UL} = T_{DL}$ for balanced DL and UL traffic loads.

Fig. 6 shows the throughput of the configurations simulated in Fig. 5. The non-overlapped design shows a constant throughput along K, proportional to the transmission rate of the MCS selected by solving (4), which yields a bitrate $R_k = 16.97$ Mbps. The overlapped windows designs initially show a rapidly growing tendency, but as K grows larger, the slope decreases until the throughput reaches a practically constant value. Interestingly, the value that the full overlap curve tends to is the throughput that would result if the transmission time of every packet was equal to its average, i.e., $s_k = \mathbb{E}\{s_k\}$. This upper bound is shown in the figure as a dashed line. As a reference, this throughput is over 3 times larger than the non-overlapped one. However, the upper bound can only be (nearly) achieved by implementing full overlap with an unreasonable amount of flows. Looking at a realistic number of flows, for instance $K = 10$, if minimum delay is desired, the throughput gain with respect to the non-overlapped case is 82%. In cases with more delay tolerance, this gain could reach up to 127%.

C. Packet loss rate simulation

In this subsection we show the effect of the packet loss rate requirement ($\breve{p}$) on the performance of the frame design. The simulation shown in Fig. 7 computes the minimum control cycle to support 10 flows with the parameters established in subsection V-3, except for the packet loss tolerance $\breve{p}$ which is now variable.

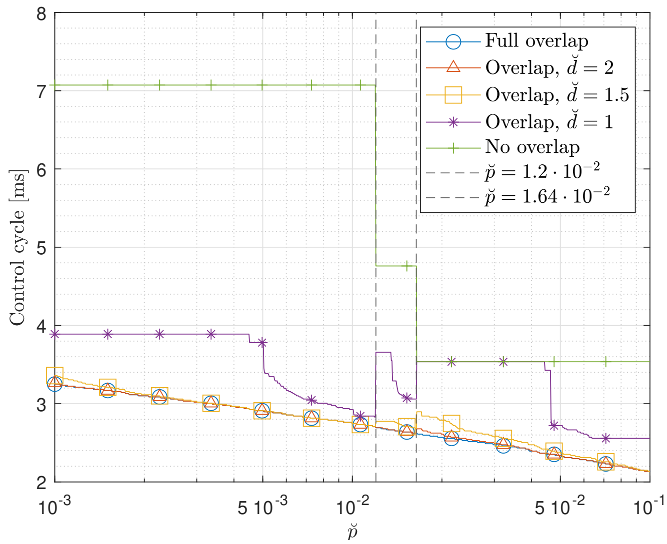

Fig. 7. Control cycle for different delay constraints. Delay constraints are normalized by the non-overlapped window length. Number of flows is $K = 10$.

Fig. 7 presents the curves for the same configurations shown in the previous simulations. The non-overlapped windows curve follows a step function, having two drops at the values of $\breve{p}$ highlighted with dashed lines in the figure. These two drops coincide with the probabilities at which it is possible to commute to a higher transmission rate R_k (i.e., higher MCS index) in (4) and thus reduce the window length N^{min}. As in the previous simulations, delay constraints $\breve{d}$ are referenced to the minimum window length N^{min}. However, as this value is not constant throughout the range of scanned probabilities, the figure is split in three separate sections, one for each N^{min} value: $N^{min} = 52$, 35, and 26 slots, respectively. In each section, the overlap configurations generally show shorter control cycles than in the non-overlapped case, and the gap widens as $\breve{p}$ increases. In particular, in the leftmost section of the figure ($\breve{p} < 1.2 \cdot 10^{-2}$), the control cycle can be roughly divided by two, even in the minimum overlap case ($\breve{d} = N^{min}$). As shown in the figure, when the delay constraint is relaxed, results are not so affected by the discretization of R_k (finite MCS table) and quickly converge to the smooth curve achieved with full overlap.

VI. Conclusion

In this paper, we have proposed a novel frame design for the wireless segment of TSN networks based on an enhanced version of the window reservation mechanism of standard 802.1Qbv. In our design, the window designed for each specific flow overlaps with those designed for other flows. The intended flow has the lowest scheduling priority in the region where the window overlaps with other windows that start before in time. On the other hand, the intended flow has the highest scheduling priority in the region that does not overlap with previous windows.

While a frame design based on non-overlapped windows that can be used only by the corresponding flow (baseline approach) can be inefficient, our approach, combined with scheduling and link control, can efficiently use radio resources. This efficiency translates into a smaller control cycle for time-sensitive applications, such as communicating control information between a central server and several wireless terminals. The control cycle reduction compared to the baseline approach is more significant as the number of terminals increases. This improvement in control cycle directly relates to a substantial increase in throughput, up to 127% specifically for 10 terminals in the case of our simulations.

References

[1] Z. Maletic, M. Mladen and M. Ljubojevic, “A Survey on the Current State of Time-sensitive Networks Standardization,” 2023 10th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN), East Sarajevo, Bosnia and Herzegovina, 2023, pp. 1-6

[2] ”IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications,” in IEEE Std 802.1AS-2020 (Revision of IEEE Std 802.1AS-2011) , vol., no., pp.1-421, 19 June 2020.

[3] ”IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks,” in IEEE Std 802.1Q-2022 (Revision of IEEE Std 802.1Q-2018) , vol., no., pp.1-2163, 22 Dec. 2022.

[4] Adame,T.; Carrascosa-Zamacois, M.; Bellalta, B. Time-Sensitive Networking in IEEE 802.11be: On the Way to Low-Latency WiFi 7. Sensors 2021, 21, 4954.

[5] “Integration of 5G with Time-Sensitive Networking for Industrial Communications”, 5G-ACIA White Paper, Jan. 2021.

[6] “Time Sensitive Networks for Flexible Manufacturing Testbed Characterization and Mapping of Converged Traffic Types”, Industrial Internet Consortium White Paper, Mar. 2019.

[7] “IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN,” in IEEE Std 802.11ax-2021 (Amendment to IEEE Std 802.11-2020), pp.1-767, May 2021.

[8] ETSI TS 123 501 V17.4.0 “5G; System architecture for the 5G System (5GS)” (3GPP TS 23.501 version 17.10.0 Release 17), Sep. 2023.

[9] E. Khorov, A. Kiryanov, A. Lyakhov and G. Bianchi, “A Tutorial on IEEE 802.11ax High Efficiency WLANs,” in IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 197-216, 2019.

[10] W. Anwar, S. Dev, K. Kulkarni, N. Franchi and G. Fettweis, “On PHY Abstraction Modeling for IEEE 802.11ax based Multi-Connectivity Networks,” 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 2019, pp. 1-7.

[11] “Physical Layer Abstraction for System-Level Simulation”. WLAN Toolbox Documentation. Matlab R2021.

[12] Ó. Seijo, Z. Fernández, I. Val and J. A. López-Fernández, “SHARP: A novel hybrid architecture for industrial wireless sensor and actuator networks,” 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), Imperia, Italy, 2018, pp. 1-10

[13] R. Croonenbroeck, L. Underberg, A. Wulf and R. Kays, ”Measurements for the development of an enhanced model for wireless channels in industrial environments,” 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Rome, Italy, 2017, pp. 1-8