R-TWT in Wi-Fi 7 and Beyond: Enabling Bounded Latency, Energy Efficiency, and Reliability

Erfan Mozaffariahrar*, Francesc Wilhelmi†, Lorenzo Galati-Giordano‡,
Pasquale Imputato§, Michael Menth*, and Stefano Avallone§
*University of Tuebingen, Germany, †Universitat Pompeu Fabra, Spain
‡Nokia Bell Labs, Germany, §University of Naples, Italy
Email: {erfan.mozaffariahrar, menth}@uni-tuebingen.de, francisco.wilhelmi@upf.edu,
lorenzo.galati_giordano@nokia-bell-labs.com, {pasquale.imputato, stavallo}@unina.it

***Abstract*—Applications with deterministic quality of service (QoS) requirements are becoming widespread, driving the evolution of wireless networks to meet strict performance demands. Wi-Fi 7, the latest generation of IEEE 802.11 standard, introduces Restricted Target Wake Time (R-TWT). It serves latency-sensitive traffic by providing contention-free channel access for certain devices and reducing their energy consumption. This paper evaluates the performance of R-TWT across diverse service types and network sizes in terms of worst-case latency, energy efficiency, collision rate, and throughput. The results, generated using the ns-3 simulator, show that R-TWT achieves bounded latency for sensitive traffic types, such as Internet of Things (IoT) and real-time (RT) applications. Moreover, R-TWT outperforms legacy mechanisms, i.e., Power Saving Mode (PSM) and Distributed Coordination Function (DCF), by achieving higher energy efficiency and a lower collision rate. Furthermore, R-TWT maintains stable and scalable performance as network size increases. This makes it a robust solution for latency-sensitive and energy-efficient wireless applications.**

***Index Terms*—R-TWT, Wi-Fi 7, latency, reliability, energy.**

I. Introduction

Every Wi-Fi generation, based on new amendments to the IEEE 802.11 standard, comes along with performance and quality of service (QoS) improvements to support new use cases with stringent requirements [1]. Modern use cases such as smart factories, robots, medical devices, extended reality (XR), and video games demand deterministic latency, reliability, energy efficiency, and high throughput [2].

To meet the demands of modern applications requiring deterministic performance, Wi-Fi 7 (IEEE 802.11be) introduces Restricted Target Wake Time (R-TWT), which is based on the Target Wake Time (TWT) mechanism from Wi-Fi 6 (IEEE 802.11ax). TWT was designed to reduce the power consumption and contention in client devices by allowing them to negotiate scheduled wake-up times with the Access Point (AP) [3]. It was primarily conceived for Internet of Things (IoT) devices with low traffic intensity and discontinued traffic but with significant power supply constraints.

Since TWT cannot guarantee deterministic performance—the wake-up times of different devices may overlap, leading to contention and additional delays—, R-TWT introduces enhanced channel access protection and resource allocation mechanisms for latency-sensitive traffic. In R-TWT, an AP allocates exclusive channel access periods that are restricted to specific stations (STAs) and predefined traffic types within the Basic Service Set (BSS), thereby reducing contention and minimizing collisions. Furthermore, R-TWT transforms the channel into a managed resource, optimizing its utilization for latency-sensitive applications while maintaining energy efficiency for devices with limited power resources. This controlled access enhances the predictability and reliability of network performance under varying traffic conditions.

This paper investigates the capabilities of R-TWT in Wi-Fi 7 and beyond and delves into the triplet of bounded latency, energy efficiency, and reliability. The key contributions of this work are summarized as follows:

- We evaluate R-TWT via simulations across key metrics, including worst-case latency, energy consumption, collision rate, and throughput for uplink UDP and TCP traffic. We compare its performance against legacy mechanisms such as Power Saving Mode (PSM) and Distributed Coordination Function (DCF).
- We analyze R-TWT's capability to support various traffic types, including Internet of Things (IoT), real-time (RT), and bulk traffic with different latency and bandwidth requirements, and assess its robustness under mixed-traffic scenarios.
- We evaluate R-TWT's scalability by evaluating its performance for various network sizes and examining its ability to maintain performance across flows.
- We identify and analyze the trade-offs in R-TWT's design, highlighting its strengths and limitations.

The rest of the paper is organized as follows. Section II describes PSM, TWT, and R-TWT. Then we discuss related works in Section III. Section IV describes the considered R-TWT implementation, simulation setup, and performance evaluation. Finally, we conclude the paper in Section V.

II. From TWT to R-TWT

In this section, we first overview the PSM mechanism. Then, we elaborate on the TWT mechanism introduced in Wi-Fi 6 and its evolution into R-TWT in Wi-Fi 7.

A. Power Saving Mode (PSM)

PSM is a power-saving mechanism from Wi-Fi that allows STAs to conserve energy by periodically switching between sleep and active modes. In the sleep mode, an STA can temporarily turn off its radio interface to save energy. During the STA sleeping periods, the AP buffers any data destined for it. In the active mode, the STA is fully awake and ready to transmit or receive data. Specifically, the STA periodically wakes up every beacon interval to receive beacon frames broadcast by the AP, which contain a Traffic Identification Map (TIM) including information about the availability of buffered data for the STA. If data is available, the STA requests and receives it from the AP, and then switches back to sleep mode until the next beacon frame. This mechanism reduces energy consumption but is less efficient in dense networks or for applications with strict QoS requirements [4].

B. Target Wake Time (TWT)

TWT was first introduced in the IEEE 802.11ah standard to meet the stringent power efficiency requirements of IoT devices operating in the sub-1 GHz frequency band. By allowing devices to schedule wake-up times for data exchange, TWT enables devices to enter idle (or sleep) modes to save energy. However, TWT in IEEE 802.11ah was primarily designed for low-power IoT devices in sparse deployments, hence not being well-suited for scenarios with high network density, diverse traffic types, or the need for QoS differentiation [5]. The IEEE 802.11ax (Wi-Fi 6) expanded TWT to support diverse use cases beyond IoT and to improve the network capacity and efficiency in dense environments. TWT enables devices to save energy by leveraging specific wake-up times for data transmission and reception.

STAs implementing TWT negotiate a TWT agreement with the AP during the association or reassociation process. This agreement defines a schedule specifying when the STAs will wake up to exchange data with the AP (TWT setup). A TWT agreement defines the Service Period (SP), which is the time window during which the STA remains awake and ready to communicate. This SP is repeated periodically at the beginning of every Wake Interval (WI) duration. Thus, the STA wakes up every WI, initiates its SP, and exchanges data with the AP. Outside this SP, the STA may go to sleep mode, potentially reducing power consumption and avoiding channel access. As a result, the proportion of time the STA is active within its WI, referred to as the duty cycle (DC), can be calculated as:

$$DC = \frac{\text{SP}}{\text{WI}} \cdot 100. \tag{1}$$

1) TWT agreements: TWT agreements are of two types, Individual TWT Agreement and Broadcast TWT Schedule (both illustrated in Fig. 1). In an Individual TWT Agreement, the AP and the STA negotiate an agreement specifying parameters such as the WI, SP, and start time. The agreement is exclusive to the STA, meaning other STAs in the network are unaware of it, so they might be active during the SP and inflict channel contention or collisions. As shown in Fig. 1, during the TWT Setup, STA 3 requests an Individual agreement and AP acknowledges the request. After the TWT duration, i.e., the time until the agreement becomes active, the first Individual

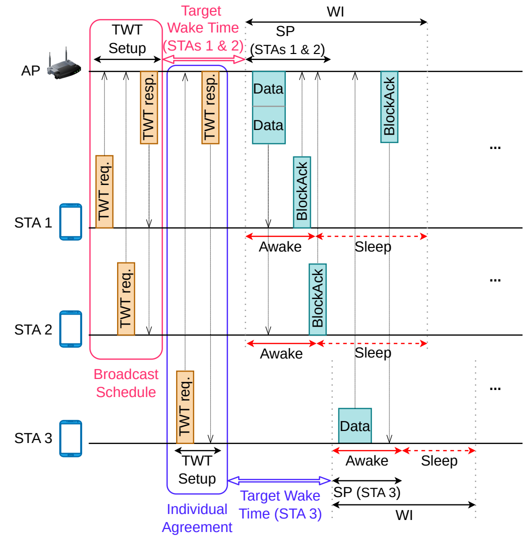


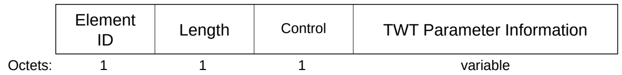
Fig. 1: Individual and Broadcast TWT Agreements.

TWT SP begins. During STA 3's SP, the STA transmits data to the AP and then receives an acknowledgment. Outside the SP, STA 3 remains in a sleep state, conserving energy.

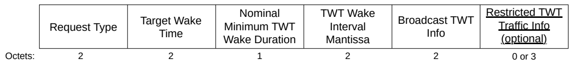
As for the Broadcast TWT Schedule, the AP broadcasts a schedule to the network, which is understood by all TWT-enabled STAs. This schedule targets a group of STAs in the network to wake up during the defined SP and contend for channel [6]. While this mechanism improves coordination among STAs, it still results in contention within the group. In Fig. 1, STAs 1 and 2 request an agreement, and the AP solicits them a Broadcast TWT Schedule. After the TWT duration, the first SP begins and the AP sends data to both STAs and receives BlockAck from both STAs. Outside the SP, both STAs alternate between wake and sleep states, conserving energy while awaiting the next SP.

A TWT agreement can operate in various modes, depending on the requirements of the network or application. These modes include implicit or explicit, announced or unannounced, and trigger-enabled or non-trigger-enabled:

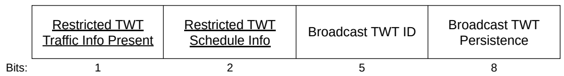
- **Implicit vs. Explicit**: While an implicit agreement occurs periodically until it is explicitly torn down, an explicit agreement requires renegotiation upon the request of the STA or AP, typically after an SP ends.
- **Announced vs. Unannounced**: In an announced agreement, the STA is expected to notify the AP its readiness to receive data. This is typically done by sending a Power Saving Poll (PS-Poll) or Automatic Power Save Delivery (APSD) trigger frame to the AP, signaling that it is awake and ready for communication. In an unannounced agreement, in contrast, the AP sends the frames to the STA without requiring any prior signal, trusting that the STA wakes up at the scheduled time.



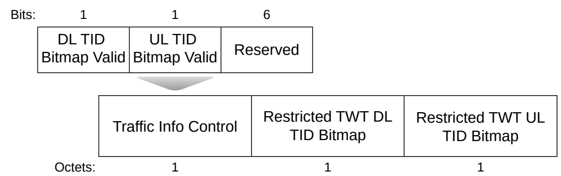
(a) TWT Element



(b) Broadcast TWT Parameter Information field



(c) Broadcast TWT Info field



(d) Restricted TWT Traffic Info field

Fig. 2: TWT Element Including Broadcast TWT Parameter Information.

- **Trigger-Enabled vs. Non-Trigger-Enabled**: In a trigger-enabled TWT agreement, the AP explicitly controls the uplink transmissions during the SP by sending trigger frames to the STA. But in non-trigger-enabled mode, STAs do not wait for any signal from the AP and contend for channel access if they have data to send.

2) TWT signaling: TWT agreements are established and managed through an Information Element (IE) known as the TWT element in Wi-Fi. A TWT element conveys the necessary parameters for scheduling SPs and managing operations. As shown in Fig. 2a, it consists of four primary fields: Element ID, Length, Control, and TWT Parameter Information. The *Element ID* identifies the type of element within the management frame, while the *Length* field specifies the size of the Control and TWT Parameter Information fields. The *Control* field defines key attributes of the agreement, including the type of TWT agreement (Individual or Broadcast), power-saving mode, and operational parameters.The *TWT Parameter Information* field contains configuration details specific to one or multiple TWT agreements. These include Request Type, flags for operational modes such as trigger-enabled/non-trigger-enabled and announced/unannounced, and timing details for SPs and WIs.

C. Restricted Target Wake Time (R-TWT)

TWT was originally proposed to reduce power consumption while providing scheduled access between the STA and AP, mitigating contention. However, TWT does not offer QoS or prioritization. To address this limitation, Wi-Fi 7 introduces R-TWT as a mechanism for defining new types of TWT agreements that prioritize latency-sensitive traffic.

R-TWT is built on top of Broadcast TWT Schedule in Wi-Fi 6 and incorporates deterministic scheduling by allowing the AP to define restricted channel access periods for specific STA(s) and traffic types within the BSS. The negotiation phase of an R-TWT schedule follows the same procedure as a Broadcast TWT Schedule but with additional fields specific to R-TWT parameters in the TWT elements. A key characteristic of R-TWT schedules is their operation in a trigger-enabled mode, where the AP explicitly controls uplink transmissions by sending trigger frames to the STAs. Fig. 2b represents the TWT element for Broadcast TWT Schedule which is enhanced to support R-TWT schedules. Underlined subfields in Figs. 2b and 2c are newly added to support R-TWT.

As shown in Fig. 2b, the R-TWT traffic information is embedded in the *Broadcast TWT Parameter Information* field. The *Broadcast TWT Info* field (Fig. 2c), part of the Broadcast TWT Parameter Information field, is used to identify the presence of R-TWT parameters in the TWT element. Two subfields—*Restricted TWT Traffic Info Present* and *Restricted TWT Schedule Info*—indicate *i)* the presence of R-TWT information in the element, *ii)* the current state of the R-TWT schedule, indicating whether it is active or idle.

For R-TWT schedules, the *Restricted TWT Traffic Info* field is included in the Broadcast TWT Info field (Fig. 2b). As shown in Fig. 2d, this field specifies the type of traffic assigned to this schedule. The *Traffic Info Control* subfield determines the validity of Traffic Identifier (TID) bitmaps for uplink and downlink traffic directions. If any of the traffic directions are valid, the next two octets represent their corresponding TID bitmaps. A value of 1 in bit position p within the TID bitmap indicates that TID p is considered as latency-sensitive traffic for the corresponding directions (uplink and/or downlink). Conversely, a value of 0 means regular traffic.

R-TWT allows the AP to define restricted access periods for specific STA(s). During the configuration of R-TWT, the AP and the scheduled STA(s) must set the R-TWT Traffic Info field (Fig. 2d) in the TWT element to specify the TID(s) that handle latency-sensitive traffic in uplink and downlink. This ensures that only assigned STA(s) and their designated traffic types participate in R-TWT SPs.

To ensure that the start time of the R-TWT SP is protected and the AP or member STA(s) gain channel access as soon as possible from the expected start time, the R-TWT-enabled STAs are required to end their Transmission Opportunities (TXOPs) before the start time of the R-TWT SP. This ensures the channel is almost idle, reducing contention and enabling predictable access for R-TWT SP members. For non-R-TWT STAs within the same BSS, the AP may setup a quiet interval to quiet them, which minimizes interference during the start time of the R-TWT SPs. However, legacy STAs are not obliged to respect the quiet interval and may interfere with the established R-TWT SPs.

Fig. 3 illustrates an example of an R-TWT schedule consisting of two R-TWT-enabled STAs. AP and STA 2 are members of an R-TWT SP. Information about the R-TWT SP is broadcast via beacon frames, ensuring that all relevant devices are aware of the schedule. Before the start of the R-TWT SP, STA 1, which is not a member of the SP, has

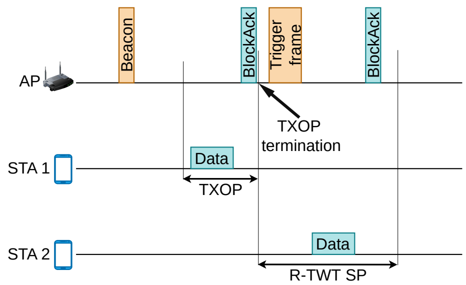


Fig. 3: Restricted TWT Schedule.

an ongoing TXOP and is exchanging data with the AP. To protect the start time of the R-TWT SP, STA 1 is required to terminate its TXOP before the SP's start time to leave the channel idle. This ensures that the AP or the STA members of the SP can gain access to the channel at that expected start time. During the R-TWT SP, the AP gains access to the channel and triggers STA 2 to transmit data using a trigger frame, ensuring contention-free access to the channel. The type of traffic transmitted by STA 2 is determined by the TID in the TWT element, which is informed via beacon frames.

This controlled mechanism significantly reduces channel contention. By restricting the access within the SP, R-TWT enables more predictable and efficient communication.

III. State of the Art

TWT and R-TWT have been conceived to address distinct objectives: TWT primarily targets optimizing energy efficiency by minimizing device power consumption, while R-TWT focuses on achieving deterministic and bounded latency for latency-sensitive traffic. Although considerable research has explored the energy-saving aspects of TWT, relatively few works have investigated the potential of TWT and R-TWT in supporting latency-sensitive traffic and enabling deterministic scheduling.

In [7], the authors propose an analytical model to estimate packet loss probability and delay probability distribution for RT applications. This model considers R-TWT parameters such as SP and WI, as well as traffic types, retransmission attempts, and network configuration. The AP utilizes this model to determine optimal R-TWT parameters, ensuring efficient support for RT applications. Haxhibeqiri *et al.* [8] implemented a cyclic TWT SP scheduling on a software-defined radio platform to emulate the R-TWT behavior. They minimize channel contention by sequentially scheduling TWT SPs and avoiding overlaps. Their experiment consists of two distinct time-critical traffic flows with unique latency bounds and shows that R-TWT guarantees latency for sensitive traffic. Similarly, [9] introduces a framework for maximizing timely throughput using Broadcast TWT schedules. The study introduces a joint optimization approach that configures TWT parameters such as SP, WI, and start time, in combination with resource allocation and STA grouping to ensure packets are delivered within their deadlines. The optimization problem of optimal TWT configuration is divided into two parts: grouping STAs and allocating resources to the members of each group. By evaluating different strategies, the paper concludes that the proposed mechanism outperforms other mechanisms for grouping and allocating resources.

Additionally, several studies have investigated methods to enhance the efficiency of TWT. For instance, traffic characterization to improve TWT scheduling is explored in [10]. The study introduces a method for accurate traffic characterization, using In-band Network Telemetry (INT) to measure traffic generation patterns accurately. It embeds traffic information in the TCP header to provide sufficient information for the AP to schedule the TWT SPs effectively. This method optimizes energy efficiency and throughput.

Some works focus on Uplink Orthogonal Frequency Division Multiple Access (UL-OFDMA) to further optimize TWT's WIs and start times agreements. For instance, the authors of [11] propose a scheduling mechanism that minimizes contention and maximizes resource allocation by managing the number of awake devices based on available Resource Units (RUs), contention window size, and WIs requested by STAs. Similarly, [12] introduces a scheme that ensures the number of awake STAs matches the available RUs at each target beacon and optimizes the WIs to achieve contention-free channel access. The authors employ a genetic algorithm to maximize throughput while satisfying delay constraints and improving energy efficiency. Authors of [13] evaluate TWT scheduling strategies by proposing a mathematical model to optimize SP duration based on STAs' traffic loads. The study compares two strategies: max-rate, which prioritizes high-rate STAs, and proportional fairness, which balances throughput and fairness. The results show that max-rate achieves higher throughput, while proportional fairness ensures better resource distribution.

The reviewed works highlight advancements in TWT and R-TWT mechanisms for energy efficiency, throughput, and latency. While substantial progress has been made in optimizing energy-saving mechanisms, research on deterministic and bounded latency remains limited, particularly for R-TWT. These gaps emphasize the need for further investigation into latency-sensitive scheduling and QoS differentiation in future wireless networks.

IV. Performance Evaluation

In this section, we first describe our implementation and outline the simulation setup used for the evaluation. We then present and analyze the obtained results.

A. Implementation Considerations

We mimic the restricted access behavior of R-TWT by adopting Individual TWT, whereby the AP negotiates TWT SPs with each STA, and by assigning non-overlapping wake-up times to specific STAs. Accordingly, only the scheduled STA is expected to communicate during the negotiated SP, hence creating a pseudo-restricted access period.

To study the QoS aspects of R-TWT, we assign a single traffic type to each individual STA, ensuring that a single traffic type is transmitted at a given TWT SP. This eliminates the need for traffic classification or internal QoS prioritization, as the STA transmits or receives only its designated traffic flow. Since the network merely consists of TWT-enabled STAs, deterministic scheduling of TWT intervals, durations, and start times, R-TWT behavior is realized, resulting in contention-free channel access for each STA.

Moreover, data packets are generated periodically but are not synchronized with the beginning of the SPs, leading to asynchronous arrivals that are buffered until the next SP. This buffering introduces queuing delays, particularly when WIs are long, as packets must wait longer for their transmission opportunity. In addition, control frames are transmitted via these R-TWT SPs, which may cause additional delays.

B. Simulation Setup

We conducted the experiments in the ns-3 simulator (version ns-3.36.1) [14] to evaluate the performance and robustness of R-TWT in handling heterogeneous application demands. Thus, we selected three traffic types, namely Internet of Things (IoT), real-time (RT), and unlimited bulk, each including distinct characteristics observed in real-world deployments:

- **IoT traffic**: which is uplink, periodic, low data rate UDP transmission common in industrial automation and monitoring. According to the Time-Sensitive Networking (TSN) testbed characterization [15], such traffic types are latency-sensitive ($\leq 20\,\mathrm{ms}$) and require deterministic delivery. R-TWT's contention-free scheduling makes it suitable for such time-sensitive applications. In our simulations, IoT flows are configured with a data rate of $500\,\mathrm{Kb/s}$, payload size of $200\,\mathrm{B}$, and a maximum latency threshold of $20\,\mathrm{ms}$.
- **Real-time (RT) traffic**: reflects the latency-critical applications such as webcam feeds, video conferencing, XR, and cloud gaming. Services like Zoom and Microsoft Teams recommend end-to-end latencies below $150\,\mathrm{ms}$, within 20–50 ms considered ideal for highly interactive experiences [16], [17]. R-TWT is designed to support such scenarios through deterministic access and bounded latency. In our setup, RT flows have a data rate of $6\,\mathrm{Mb/s}$ and are evaluated in uplink UDP and TCP. Each packet has a $1000\,\mathrm{B}$ payload, and the latency threshold for ideal user experience is set to $50\,\mathrm{ms}$.
- **Bulk traffic**: represents high-throughput, best-effort applications, such as file transfers, which saturate the channel and create contention. Including such flows allows us to assess the robustness of R-TWT under background load and its ability to isolate latency-critical traffic. Bulk traffic is modeled as uplink TCP flows with no rate limitation, aiming to utilize the full available bandwidth. Each packet has a payload size of $1300\,\mathrm{B}$.

These traffic types were chosen to emulate a realistic multi-service environment and to stress the R-TWT mechanism under diverse performance requirements. We focus exclusively on uplink traffic to evaluate how R-TWT coordinates channel access under high STA-side contention.

All experiments were conducted in a single BSS, consisting of one AP and multiple STAs, positioned 1-2 meters from the AP. The channel width is set to 40 MHz, operating in the 6 GHz frequency band, and the selected Modulation and Coding Scheme (MCS) is 7. Moreover, we use the Medium Access Control (MAC) queue size of 500 packets for the AP and the STAs.

Each experiment was performed independently, with all STAs in the network implementing a single mechanism, namely R-TWT, PSM, or DCF, which is the legacy contention-based channel access method. This separation ensures that no mixed-mechanism scenarios occur, allowing for a direct and fair comparison of the performance of each mechanism. Simulation results were averaged over five runs with different random seeds. Each simulation lasted 15 seconds plus a 1.2-second warmup phase. During the warmup phase, TWT agreements were established, and data collection began after this phase. The TWT agreements are configured to operate in implicit, unannounced, and trigger-enabled modes.

Performance evaluation focuses on the following key metrics: application-layer worst-case latency (99th percentile, in ms), energy consumption (J), collision rate ($\%$), and throughput (Mb/s). Collision rate represents the proportion of packets dropped due to overlapping transmissions at the physical layer. We classify a packet as collided if the preamble or signal fields (e.g., legacy SIG, HT-SIG, HE-SIG) are corrupted, or if reception is aborted due to interference from a new transmission. Drops caused by power state changes (e.g., sleep) or unsupported configurations are excluded from the collision count.

C. Experiment 1: R-TWT Parameter Configuration

We start showing how critical it is to properly configure the parameters of the R-TWT mechanism. In particular, a shorter WI results in more frequent SPs, which reduces latency as the device has more opportunities to access the channel. Therefore, the WI duration for each flow must be selected based on the latency bounds of the traffic to be supported. As for the SP duration, a longer value allows exchanging more data (active periods are longer), potentially reducing latency. However, this can increase the waiting time for other devices, as their transmissions are deferred until the current SP ends. It may also lead to a buffer overflow on the sender side.

To study the trade-off between reducing latency for one device and increasing it for others, we analyze the sensitivity of the RT flow to different DC values in the presence of IoT and bulk flows. Based on these results, we select the most suitable DC configuration for the RT flow and use it in the subsequent experiments that evaluate R-TWT's performance under varying network sizes. For simplicity and fairness, a common WI is used for all flows. This value is chosen based on the latency requirement of the IoT flow, which has the most stringent constraint. As a result, the WI must be short enough to ensure that the latency threshold is met.

The simulation consists of a single BSS with one AP and 11 STAs. The first STA generates an uplink UDP IoT flow, the second carries an uplink UDP RT flow, and the remaining 9 STAs generate uplink TCP bulk flows. The bulk flows serve as background traffic to inflict constant channel contention.

A fixed WI of 17408 μs is used for all flows. The IoT flow is assigned a pre-validated DC of 10% to satisfy its latency constraints. The remaining DC is split between the RT and bulk flows, such that the RT flow's DC is reciprocal to that of the bulk flows. We vary the RT DC from 6% to 18% by adjusting their SPs to evaluate the impact on key performance metrics.

Fig. 4 presents the performance of the RT flow as a function of the DC. The performance of the IoT flow, which assumes a fixed DC of 10% in all the cases, is also included. As shown in Fig. 4a, a lower DC, which results in insufficient airtime, leads to unacceptable latency for RT compared to the RT threshold. As DC increases, RT's latency decreases due to more frequent and longer SPs, thus allowing it to meet the latency requirement. However, higher DC values also lead to increased energy consumption, as shown in Fig. 4b, since the RT STA remains active for longer durations within each WI.

The collision rate remains nearly unchanged across different DCs. Similarly, the throughput of the RT flow matches its configured rate, due to the nature of UDP traffic, which continuously transmits packets without flow control. The corresponding plots are omitted to save space. Finally, allocating more DC to the RT flow reduces the airtime available to background bulk flows, leading to a decline in their aggregated throughput, as shown in Fig. 4c. Hence, the optimal DC value for UDP RT traffic is 12%.

These results highlight the importance of carefully selecting the DC in R-TWT to balance latency, energy, and fairness. To validate this behavior for different transport protocols, we repeated the experiment for the RT flow using TCP (not included for the sake of space). We determined that the optimal DC is 16% to satisfy the required performance constraints.

D. Experiment 2: Robustness of R-TWT for Various Services

In this experiment, we evaluate the robustness of R-TWT in supporting latency-sensitive traffic under mixed-traffic conditions. Specifically, we assess how IoT and RT traffic perform in the presence of uplink TCP bulk flows creating increasing channel contention. The goal is to determine whether R-TWT can maintain acceptable latency and throughput for prioritized flows, while exploring how traffic characteristics and network size influence overall network performance.

TABLE I: Configuration of traffic flows.

Flow type	***Data rate***	***Payload (B)***	***DC (%)***	***WI (μs)***
IoT	500 Kb/s	200	10	17408
Real-time	6 Mb/s	1000	12	17408
Bulk	Unlimited	1300	75	17408

The simulations consist of a single BSS with one AP and n STAs, where n ranges from 3 to 15. We consider IoT, RT, and bulk traffic types, each with distinct characteristics summarized in Table I. Throughout all the simulation scenarios, the first STA is configured with the IoT flow, the second STA carries the RT flow, and STAs 3 to n each generate a single uplink TCP bulk flow. The R-TWT configurations in this experiment are based on the empirical tests in Section IV-C.

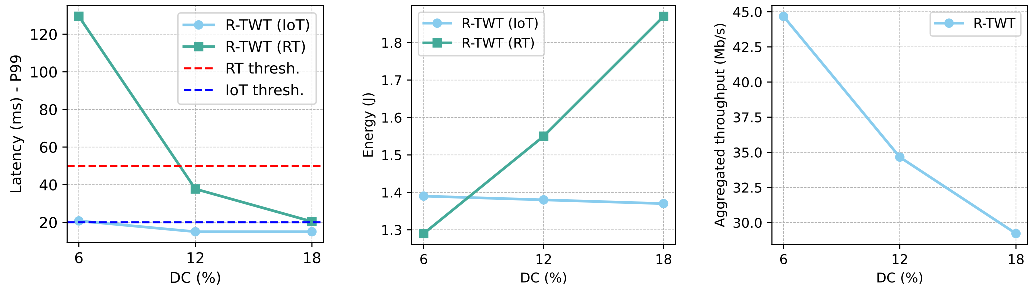
1) RT UDP: Fig. 5 evaluates R-TWT performance in terms of worst-case (99-percentile) latency (with latency thresholds of 20 ms for IoT and 50 ms for RT), energy consumption, and collision rate of IoT and RT flows under R-TWT, PSM, and DCF mechanisms. As shown in Fig. 5a, only R-TWT consistently meets both thresholds as network size increases, due to R-TWT's contention-free scheduling, which isolates traffic in predefined SPs. In contrast, PSM exceeds these thresholds in larger networks due to sleep-wake buffering and increased contention upon waking (e.g., 100 ms). Likewise, DCF leads to significant latencies (e.g., over 300 ms from 7 STAs) due to excessive channel contention and collisions.

In terms of energy consumption, R-TWT significantly outperforms both PSM and DCF (Fig. 5b), even as the network size increases, thanks to the long sleep durations that IoT and RT STAs maintain under R-TWT. The scheduled nature of R-TWT also reduces collisions for latency-sensitive flows, as shown in Fig. 5c. Even as the network size increases, collision rates for IoT and RT flows remain minimal in R-TWT compared to PSM and DCF. This improvement results from R-TWT's contention-free access, which isolates each flow in its designated SP.

Fig. 6 presents the throughput of IoT and RT flows, along with the aggregated throughput of background bulk transmissions. As for the throughput (Fig. 6a), both IoT and RT flows achieve their defined targets (500 Kb/s for IoT, 6 Mb/s for RT). This shows that R-TWT maintains predictable throughput for latency-sensitive traffic across varying network sizes. However, this comes at the cost of overall network performance. As shown in Fig. 6b, R-TWT significantly reduces the aggregated throughput of bulk flows compared to PSM and DCF, due to the reduced channel access time allocated to bulk transmissions. This highlights R-TWT's design priority on deterministic access, which limits bandwidth available for best-effort traffic in favor of protecting latency-sensitive flows.

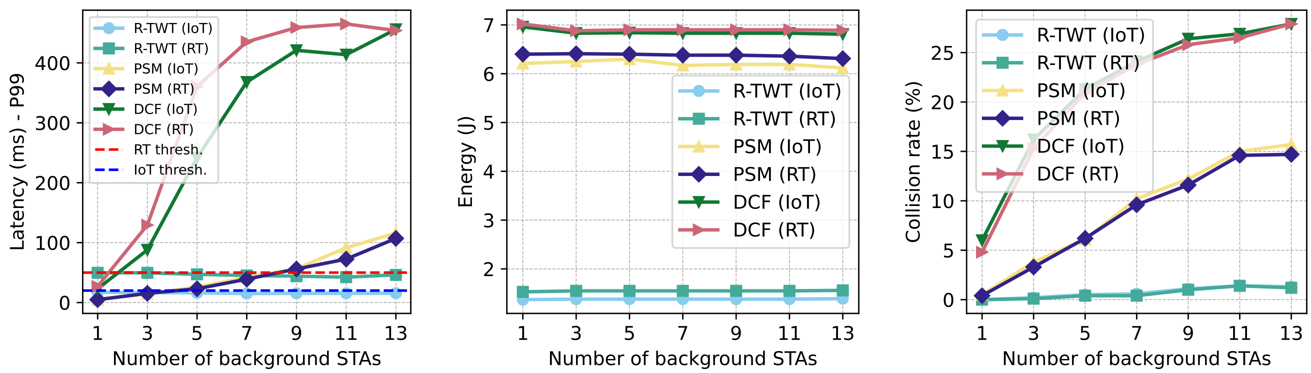
2) RT TCP: Changing the transport protocol of the RT flow to TCP results in similar trends in energy consumption and collision rate. Therefore, we omit those plots due to space constraints. As motivated in Section IV-C, fixed DCs of 10% and 16% are chosen for IoT and RT flows, respectively. The remaining DC is reserved for bulk flows.

Fig. 7a shows the worst-case uplink latency for IoT UDP and RT TCP flows under R-TWT, PSM, and DCF as the number of background STAs increases. R-TWT maintains stable and bounded latency, consistently meeting the latency thresholds for both flows (20 ms for IoT, 50 ms for RT), even as network density increases. This demonstrates R-TWT's ability to provide contention-free access through isolated SPs. PSM shows relatively stable latency growth but exceeds the 50 ms threshold at larger network sizes (e.g., 10 STAs). This



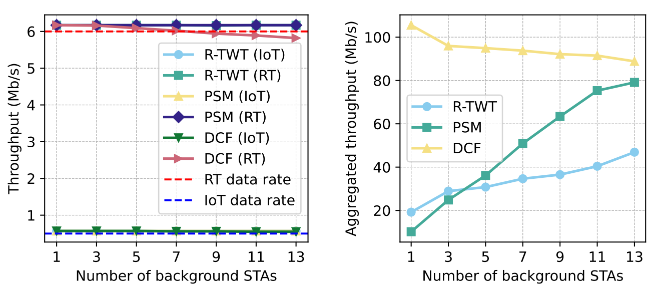
(a) Worst-case latency (b) Energy consumption (c) Aggregated bulk throughput

Fig. 4: Uplink UDP real-time (RT) traffic with DC varied from 6% to 18%; IoT flow fixed at 10%.



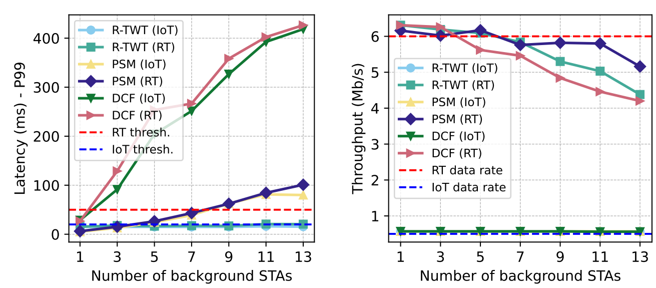
(a) Worst-case latency (b) Energy consumption (c) Collision rate

Fig. 5: Uplink IoT and real-time (RT) UDP traffic performance under R-TWT, PSM, and DCF, across key metrics.



(a) Throughput (b) Aggregated bulk throughput

Fig. 6: Throughput of uplink IoT and RT UDP traffic.



(a) Worst-case latency (b) Throughput

Fig. 7: Uplink IoT UDP and RT TCP traffic performance.

degradation is due to increased buffering at the AP and delayed channel access caused by sleep-wake transitions. In contrast, DCF exhibits a sharp rise in latency for both traffic types, exceeding 300 ms for 8 STAs. This behavior is the result of contention and frequent retransmissions.

As for throughput shown in Fig. 7b, the IoT flow maintains its expected throughput across all mechanisms. This is due to its low data rate and the ability of R-TWT, PSM, and DCF to accommodate such traffic even under increasing contention.

In contrast, the RT TCP flow under R-TWT experiences a gradual throughput degradation as the network grows. This is due to the buffer size at the AP. As more uplink TCP flows are introduced, the AP must manage an increasing volume of TCP acknowledgments. Given limited buffer capacity, acknowledgment packets for the RT flow are dropped. The STA interprets these missing acknowledgments as congestion, resulting in reduced congestion window size and a reduced send rate. Consequently, the STA cannot fully utilize its allocated SPs, leading to lower throughput.

In PSM, the RT flow also suffers throughput degradation due to delayed acknowledgment delivery and increased channel access contention during active periods. DCF, on the other hand, starts with higher throughput in low-density scenarios but quickly degrades as contention and collisions increase with network size.

3) Takeaways: The results confirm that R-TWT effectively provides low-latency guarantees for latency-sensitive traffic, such as IoT and RT applications, while maintaining energy efficiency and consistent per-flow throughput and reliable channel access across varying network sizes. This is primarily because of R-TWT's contention-free scheduling mechanism, where specific STAs and traffic types are assigned to predefined SPs, ensuring timely and isolated access. As a result, R-TWT significantly reduces collision rates for prioritized flows, further enhancing reliability.

However, these benefits come with trade-offs. Bulk flows experience degraded performance due to limited airtime under R-TWT's fixed schedule. Additionally, TCP-based RT flows suffer throughput degradation in larger networks because of acknowledgment packet drops at the AP caused by buffer overflow. These dropped acknowledgments disrupt the TCP behavior, reducing congestion window growth and underutilizing the allocated SPs. While R-TWT provides strict timing guarantees, it does not manage or guarantee buffer space at the receiver. As a result, buffer limit, particularly at the AP, can significantly affect end-to-end TCP performance.

Finally, in dense deployments, the requirement to guarantee contention-free access for many STAs reduces the AP's flexibility in allocating sufficient SP durations per device. As a result, the limited airtime per flow can lead to increased latency and reduced throughput.

V. Conclusion

In this paper, we modeled and evaluated the behavior of R-TWT in the ns-3 simulator to assess its potential at enhancing low latency, energy efficiency, and reliability. Through simulations, we assessed the robustness of R-TWT against diverse traffic types and showcased the scalability of the mechanism for varying network sizes. The results highlight that R-TWT achieves bounded latency for latency-sensitive traffic, such as IoT and RT applications, meeting the expected bounds for these use cases. Furthermore, R-TWT scales effectively with network size, maintaining consistent throughput and achieving superior energy efficiency in all scenarios. R-TWT outperforms PSM and DCF in most of the evaluated metrics, particularly for dense environments where contention is high. Moreover, R-TWT interacts better with UDP applications such as industrial sensing and surveillance. It may not be ideal for TCP-based flows, as its rigid scheduling can lead to buffer overflow and reduced throughput due to disrupted congestion window.

In future work, we plan to evaluate R-TWT performance under more congested and heterogeneous traffic conditions to further validate its scalability in dense deployments. We also aim to investigate adaptive DC configurations to enhance channel access time and efficiency in dynamic network environments. Finally, we will explore the impact of buffer management in R-TWT scheduling on TCP-based flows.

Acknowledgment

F. Wilhelmi is partially funded by Wi-XR PID2021-123995NB-I00 (MCIU/AEI/FEDER,UE). L. Galati-Giordano is partially supported by UNITY-6G project 101192650 (EU/SNS JU).

References

[1] L. Galati-Giordano, G. Geraci, M. Carrascosa, and B. Bellalta, "What will Wi-Fi 8 be? A Primer on IEEE 802.11 bn Ultra High Reliability," *IEEE Communications Magazine (COMMAG)*, 2024.

[2] D. Cavalcanti, C. Cordeiro, M. Smith, and A. Regev, "WiFi TSN: Enabling Deterministic Wireless Connectivity over 802.11," *IEEE Communications Standards Magazine*, 2022.

[3] E. Mozaffariahrar, F. Theoleyre, and M. Menth, "A survey of Wi-Fi 6: Technologies, Advances, and Challenges," *MDPI Future Internet Journal (FI)*, 2022.

[4] M. N. Alam, R. Jäntti, JarkkoKneckt, and J. Nieminen, "Performance Analysis of the IEEE 802.11 s PSM," *Journal of Computer Networks and Communications*, 2012.

[5] B. Domazetović, E. Kočan, and A. Mihovska, "Performance Evaluation of IEEE 802.11 ah Systems," in *Telecommunications Forum (TELFOR)*. IEEE, 2016.

[6] IEEE, "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," *IEEE Std 802.11ax-2021 (Amendment to IEEE Std 802.11–2020)*, 2021.

[7] A. Belogaev, X. Shen, C. Pan, X. Jiang, C. Blondia, and J. Famaey, "Dedicated Restricted Target Wake Time for Real-Time Applications in Wi-Fi 7," *arXiv preprint arXiv:2402.15900*, 2024.

[8] J. Haxhibeqiri, X. Jiao, X. Shen, C. Pan, X. Jiang, J. Hoebeke, and I. Moerman, "Coordinated SR and Restricted TWT for Time Sensitive Applications in WiFi 7 Networks," *IEEE Communications Magazine (COMMAG)*, 2024.

[9] R. Roy, R. V. Bhat, P. Hathi, N. Akhtar, and N. M. Balasubramanya, "Maximization of Timely Throughput with Target Wake Time in IEEE 802.11 ax," in *IEEE International Conference on Communications (ICC)*. IEEE, 2023.

[10] J. Sheth, V. K. Ramanna, and B. Dezfouli, "Traffic Characterization for Efficient TWT Scheduling in 802.11 ax IoT Networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2023.

[11] Q. Chen, G. Liang, and Z. Weng, "A Target Wake Time based Power Conservation Scheme for Maximizing Throughput in IEEE 802.11 ax WLANs," in *International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2019.

[12] Q. Chen and Y.-H. Zhu, "Scheduling Channel Access based on Target Wake Time Mechanism in 802.11 ax WLANs," *IEEE Transactions on Wireless Communications*, 2020.

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

[14] E. Mozaffariahrar, M. Menth, and S. Avallone, "Implementation and Evaluation of IEEE 802.11 ax Target Wake Time Feature in ns-3," in *Proceedings of the 2024 Workshop on ns-3*, 2024.

[15] Industrial Internet Consortium, "Characterization and Mapping of Converged Traffic Types: Time Sensitive Networking (TSN) Testbed," Industrial Internet Consortium, Tech. Rep., 2019. [Online]. Available: https://www.iiconsortium.org/pdf/IIC_TSN_Testbed_Char_Mapping_of_Converged_Traffic_Types_Whitepaper_20180328.pdf

[16] Zoom Video Communications., "System Requirements for Zoom Rooms," 2025. [Online]. Available: https://support.zoom.com/hc/en/article?id=zm_kb&sysparm_article=KB0060748

[17] Microsoft Corporation, "Media Quality and Network Connectivity Performance in Skype for Business," 2017. [Online]. Available: https://learn.microsoft.com/en-us/previous-versions/SkypeForBusiness/optimizing-your-network/media-quality-and-network-connectivity-performance